Shape optimization using feature-based CAD systems and adjoint methods

Dheeraj Agarwal, MEng

June 2018

School of Mechanical and Aerospace Engineering

Queen's University Belfast

A thesis submitted for the degree of

Doctor of Philosophy

This thesis is dedicated to the memory of my father

Mukesh Kumar Agarwal

(02/06/1956 – 17/05/2012)

Abstract

The key issue restricting the use of computer-aided design (CAD) models within an optimization framework is that there is no clear definition of how the change in CAD parameters effect the model's performance in terms of optimizing for certain objective functions (for e.g. minimum pressure loss, minimum drag, maximum lift etc.). In this thesis, an automated optimization process is presented, which uses the parameters defining the features in a feature-based CAD model as design variables. This process exploits adjoint methods for the computation of gradients, which predicts how the objective function changes for an infinitesimally small movement of each surface mesh node in the normal direction. The use of adjoint methods results in a computational cost that is essentially independent of the number of design variables, making it ideal for optimization in a large parameter space.

The success of any shape optimization methodology depends on the choice of parameters and can sometimes stifle the creation of high performing innovative solutions. Parametric effectiveness is a measure that rates the ability of the parameters in a model to change its shape in the optimum way. Here, the optimum shape change is that suggested by the adjoint sensitivity on the model boundary. Herein, an automated approach is developed to compute the parametric effectiveness of CAD model parameters. In cases where the parametric effectiveness is low, a novel methodology is shown which automatically adds the optimum features to the CAD model feature tree, and thus increases the design freedom of the model. In this thesis, the optimization framework is developed to exploit the capabilities of modern CAD systems to add geometrical constraints to the optimization process including minimum thickness, constant volume and packaging constraints. The packaging constraints are imposed by the adjacent components in the CAD model product assembly which the component being optimized is not allowed to violate.

The applicability of the developed approaches is demonstrated on a range of CAD models created in CATIA V5 for 2D and 3D finite element and computational fluid dynamics problems. During this research, the ability to carry out optimization directly on the CAD models created in commercial CAD systems has been enhanced. In addition, it has been shown that the additional shape flexibility imparted to the model by inserting additional "optimum" CAD features, leads to a better optimized

component than would have been possible using the original model. Lastly, it has been shown that an optimization process can be configured to respect CAD assembly constraints, resulting in an optimized geometry that does not violate the space occupied by other components in the product assembly.

Acknowledgements

First and foremost, I wish to thank my supervisors Dr. Trevor T. Robinson and Prof. Cecil G. Armstrong for giving me the opportunity to work under their expert guidance. They have been a source of continuous motivation throughout this work, and I am indebted to them for providing an excellent atmosphere to carry out my research. It has been a humbling experience working with them, to say the least.

I also wish to acknowledge the financial support by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

I'd like to extend my sincere appreciation to all the members of the Finite Element Modelling Group at Queen's University Belfast for providing support and creating an enjoyable office atmosphere.

Finally, I am very grateful to my family in India, this submission would not have been possible without their extraordinary care and support. Last but not the least I would like to pay special thanks to my wife Mrs. Richa Agarwal for her unconditional support and understanding.

Table of Contents

List of Figures

Nomenclature

J	objective function
$\boldsymbol{X}$	mesh coordinates
$\boldsymbol{U}$	vector of the fluid system variables
$\boldsymbol{R}$	residual of flow solutions
$\boldsymbol{\psi}$	adjoint solution
u, v, w	parametric coordinates
i, j, k	control point
$P_{i,j,k}$	coordinates of the control point
D	Bernstein polynomials
$B_{i,j}$	basis functions
FFD	Free-form deformation
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CFD	Computational fluid dynamics
FEA	Finite element analysis
SLSQP	Sequential least square programming
θ	Design parameters
B-Rep	Boundary representations
AD	automatic differentiation
API	application programming interface
V_n	Design velocities
$\hat{\boldsymbol{n}}$	direction of surface normal
dA	facet area
ζ, η, ξ	Barycentric coordinates
φ_i	twist
ϕ	adjoint sensitivity
g, h	constraint functions
PS	Pressure side
SS	Suction side

LE	Leading edge
TE	Trailing edge
RANS	Reynolds-averaged Navier-Stokes
SU2	Stanford University unstructured
C_l	Lift coefficient
C_d	Drag coefficient
C_p	Pressure coefficient
α	angle of attack
c	chord length
l	lift
d	drag
CAA	Component Application Architecture
RADE	Rapid Application Development Environment
CFL	Courant-Friedrichs-Lewy number
QUB	Queen's University Belfast
RRD	Rolls-Royce Deutschland
VKI	Von Karman Institute for Fluid Dynamics
VW	Volkswagen Group Research

Chapter 1 Introduction

With advances in the field of computers and their progressive use within the industrial design process, the need for physical design prototypes have been extensively reduced and replaced with that for digital models which are constructed and analyzed using computers. Nowadays product design typically starts with the construction of a CAD geometry of an initial concept and the goal is to deliver the geometry (optimized to meet the objective function) as a CAD model which can be used for manufacturing. However, currently there is no efficient way of either optimizing directly on feature-based CAD models, or of generating a CAD model from optimization performed on computer-aided engineering (CAE) meshes. The current techniques to capture the geometry from the optimized meshes require extensive effort, are time consuming and lose important geometric details. Hence, there is a significant advantage to be realized by directly using the CAD models within the optimization framework.

Current research in this area aims to enable shape optimization by using either a "dumb" geometry, which is a non-parametric CAD model from which the construction history has been removed or using a feature-based CAD model with its construction history, features and parameters included. In general, all the commercial CAD systems like CATIA V5 [1], SIEMENS NX [2], SolidWorks [3] etc. use feature based modelling strategies to create a parametric CAD model. This capability of modern CAD systems enables the designer to create relationships between different CAD features, and sometimes between different parts or assemblies, to integrate the design intent for the model. The main advantage of the parametric approach is that the optimized model can be directly used for downstream applications including manufacturing and process planning. However, the main disadvantage of optimizing the CAD model is that the final design will just be a parametric variation of the initial one, rather than a radically different shape. Also, there will be situations where the

parameters associated with the features used for CAD model design may not be the best choice for optimization. In these cases, the design will never reach a true optimum and to get around with this issue it becomes essential to either re-parameterize the existing features or add more features to the CAD model to increase its flexibility.

In recent years, optimization has become an essential and integrated part of the industrial design process. The need for optimizing designs to a global optimum led to the development of various stochastic methods like genetic and evolutionary algorithms [4, 5]. One of the limitations of these methods is the requirement for many function evaluations to converge to an optimal solution. In the field of computational fluid dynamics (CFD), a typical runtime for an industrial component ranges from hours to days on high performance clusters [6], thus using stochastic methods for routine design becomes prohibitive. In this regard, the use of a gradient-based optimization methods which requires very few iterations to reach an optimum is desirable. However, this requires the gradient of the objective function with respect to the design variables to be computed. One of the straight-forward way to get the gradients for each design variable is to employ a finite difference technique, where the effect of a parameter change is computed by analysing the performance of both the baseline and perturbed designs and comparing the results. For a typical CAD model, this requires a perturbed geometry to be created for each parameter in the CAD system and then used for analysis (including the need for geometry healing, application of properties and boundary conditions, and mesh generation processes), where the resulting difference in performance enables the derivative calculation.

There is a desire to use CAD model parameters as design variables for optimization, but one of the key issues restricting this ambition is that there is no clear link between the CAD parameters and how these parameters effect the model's performance. Also, the successful integration of a CAD model in a gradient based optimization loop requires an efficient way of calculating the gradients of the objective function with respect to the CAD parameters. Robinson et al. [7] has shown an approach which enables this by linking adjoint sensitivities with the parametric design velocity, i.e. the boundary shape movement resulting from a change in a CAD parameter. Adjoint methods enable the computation of adjoint sensitivities which gives the information about how the objective function changes for an infinitesimally small movement of each surface mesh node in the normal direction, and have been extensively researched

in last two decades [8-16]. The primary attraction of adjoint methods is their ability to compute gradient information at a computational cost which is essentially independent of the number of design variables. This, in turn, opens the possibility to explore significantly larger design spaces than those possible with traditional approaches, in time-scales which are acceptable for industrial design.

Optimization processes are not only driven by performance but are also subjected to constraints. One of these constraints is imposed by the availability of packaging space in the final assembly design where the optimized component is expected to fit, which is typically defined by other components in the assembly. Since, different components are designed and optimized by different engineers, when the components are assembled together, issues such as fit often occur, requiring engineering changes late in the product development cycle [17]. Thus, it is important for designers and manufacturers to develop methods to ascertain that the designed components can be assembled before the actual component is manufactured.

The research presented in this thesis is supported by Marie Skłodowska-Curie actions for Horizon 2020 project IODA [18], which stands for Industrial optimal design using adjoint CFD. IODA follows on from the EC projects FlowHead [19] and AboutFlow [20]. The aim is to advance with the systematic integration of adjoint-based design optimization with CFD into the regular industrial development processes. The research herein aims to contribute to this aim by developing methodologies to facilitate the use of commercial CAD systems within an industrial optimization workflow.

This research has previously been published as:

1. Agarwal D., Robinson T.T., Armstrong C.G., Kapellos C., CAD-based optimization using Adjoint methods by automatically updating the parameterization of sketch-based features (2018), submitted to Structural and Multidisciplinary optimization.

2. Agarwal D., Kapellos C., Robinson T.T., Armstrong C.G., Using parametric effectiveness for efficient CAD-Based adjoint optimization (2018), Accepted to Computer-Aided design and applications journal.

3. Agarwal D., Robinson T.T., Armstrong C.G., Marques S., Vasilopoulos I., Meyer M., Parametric design velocity computation for CAD-based design

optimization using adjoint methods (2018), Engineering with Computers, 34(2):225-239.

4. D. Agarwal, S. Marques, T. T. Robinson, P. Hewitt and C. G. Armstrong, Aerodynamic shape optimization using feature-based CAD systems and adjoint methods (2017), at 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.

1.1 Thesis outline

The outline of the remainder of this thesis is structured as follows:

Chapter 2 reviews the existing methodologies in the field of CAD-based optimization methods.

Chapter 3 presents a robust methodology for the computation of design velocities for parameters defined in a feature-based CAD model created in commercial CAD systems. The design velocities are then linked with adjoint sensitivities to compute the performance gradients and validated against the finite difference results.

Chapter 4 uses methodologies developed in chapter 3 to perform CAD-based optimization for a series of aerodynamic test cases with increasing complexity.

Chapter 5 presents an automated approach to rate the quality of CAD model parameters to be used for optimization. It also outlines a novel methodology to select the optimum set of parameters which would provide the greatest potential for performance improvement.

Chapter 6 describes a novel methodology to increase the design flexibility of a CAD model by inserting new features into the CAD model feature tree.

Chapter 7 presents an optimization process, which uses adjacent components in the CAD model product assembly to enforce constraints on the design space.

Chapter 8 presents the overall discussion of the work in this thesis.

Chapter 9 concludes the research done in this thesis.

Chapter 10 outlines the areas of future research.

Chapter 2 Literature review

2.1 Optimization methods

In general, optimization is defined as the minimization of a chosen objective function through the manipulation of a set of design variables. A general optimization framework can be formulated as:

$$\begin{aligned} \textit{Minimize } \theta\text{:} \quad & J(\theta), \\ \textit{Subject to}\text{:} \quad & g(\theta) > 0, \\ & h(\theta) = 0 \end{aligned}$$

where $J(\theta)$ is the objective function to be minimized (maximized), $g(\theta)$ are the inequality constraints and $h(\theta)$ represent equality constraints. Numerous optimization methods have been developed to date and can mainly be classified in two categories: stochastic methods and gradient based methods.

Stochastic optimization can be described as a process in which the objective function is optimized in presence of randomness. The most common stochastic methods are the population-based approaches like genetic and evolutionary algorithms [4, 5]. In this approach, the first step is the selection of initial population of designs, generated using random values of design variables. In the next step, new designs are created from existing designs by applying the principle of crossover and mutation. The newly created designs are then analyzed and ranked based on the objective function and the available constraints. The worst performing designs are then removed from the population and the procedure is repeated till a convergence is achieved.

In gradient-based optimization methods, the gradients of objective functions are evaluated and subsequently used to drive the optimization towards an optimum. Here, the first step is the computation of gradients, which defines the search direction. The second step is performing a line-search operation to obtain a suitable step-size to move

in the search direction to minimize (maximize) the objective function. Gradient descent or steepest descent is the most commonly used gradient-based optimization algorithm. It minimizes the objective function $J(\theta)$ by perturbing the parameters proportional to their individual gradients of objective function (∇J). Though this algorithm is easy to implement, it usually suffers from a slow convergence rate. There are several other optimization algorithms like Broyden–Fletcher–Goldfarb–Shanno (BFGS), Sequential least square programming (SLSQP) etc. which are highly efficient in the line-search process resulting in faster convergence.

As outlined by Zingg et. al. [21], the limitations of gradient-based approaches are the strengths of genetic algorithms and vice-versa, and the choice of optimization algorithm is problem dependent. One of the key advantages of gradient-based approaches over genetic-algorithms is their efficiency for optimizing designs with a large number of design variables. Moreover, a clear definition of convergence criterion and speed of convergence are other benefits associated with the gradient-based methods. In genetic algorithms, the main benefit arises from the fact that the probability to obtain a global minimum with a smaller design space is higher (not guaranteed) compared to gradient-based methods. The main limitation of genetic-algorithms is the high computational cost associated with the large number of function evaluations required to reach the optimum. Efforts have also been made to develop hybrid optimization methods employing the use of genetic algorithms along with gradient-based methods [22, 23].

In this thesis, gradient based optimization methods are used for the problem of shape optimization. This requires an efficient methodology for the computation of the gradient of an objective function as well as the constraints (if any). The most straight forward route of calculating the gradients is by employing finite differences, which is based on the approximation of the Taylor-series expansion as

$$J(\theta + d\theta) = J(\theta) + d\theta \left(\frac{\partial J}{\partial \theta}\right)_i + \frac{(d\theta)^2}{2}\left(\frac{\partial^2 J}{\partial \theta^2}\right)_i + \frac{(d\theta)^3}{6}\left(\frac{\partial^3 J}{\partial \theta^3}\right)_i + \cdots \qquad (2.1)$$

The gradient can be approximated by computing the objective function $J(\theta)$ for the base geometry, and $J(\theta + d\theta)$ or $J(\theta - d\theta)$ for the geometries where a perturbation of $d\theta$ has been applied in positive and negative directions respectively, known as the forward finite differences and backward finite differences respectively, Figure 2.1.

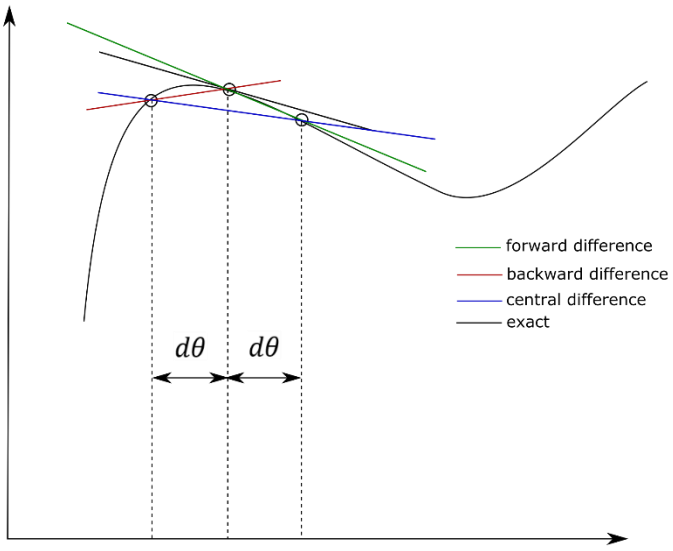


Figure 2.1 Illustration of finite difference method

The last terms in the right-hand side of the equations (2.2 – 2.4) are the errors introduced by terminating the Taylor series and are referred as the truncation errors. Thus, for finite value of higher order derivative terms ($\partial^2 J/\partial\theta^2$, $\partial^3 J/\partial\theta^3$,…) the truncation error is $\sim O(d\theta)$ for forward/backward finite differences and $\sim O(d\theta^2)$ for central finite differences.

$$\frac{\partial J}{\partial \theta} \approx \frac{J(\theta + d\theta) - J(\theta)}{d\theta} - \frac{d\theta}{2}\left(\frac{\partial^2 J}{\partial \theta^2}\right)_i - \frac{(d\theta)^2}{6}\left(\frac{\partial^3 J}{\partial \theta^3}\right)_i + \cdots, \tag{2.2}$$

$$\frac{\partial J}{\partial \theta} \approx \frac{J(\theta) - J(\theta - d\theta)}{d\theta} + \frac{d\theta}{2}\left(\frac{\partial^2 J}{\partial \theta^2}\right)_i - \frac{(d\theta)^2}{6}\left(\frac{\partial^3 J}{\partial \theta^3}\right)_i + \cdots, \tag{2.3}$$

$$\frac{\partial J}{\partial \theta} \approx \frac{J(\theta + d\theta) - J(\theta - d\theta)}{2d\theta} - \frac{(d\theta)^2}{6}\left(\frac{\partial^3 J}{\partial \theta^3}\right)_i + \cdots. \tag{2.4}$$

The benefit of the finite differences approach is that it is simple and straightforward to implement but is limited by the associated computational cost of computing additional function values which scales with the number of design variables. If the number of design variables is n, then the total number of analyses required is either $n + 1$ or $2n$ depending on whether forward/backward or central finite differences are used. The other drawback is that the accuracy is heavily dependent on the step size, which is difficult to choose a priori, Figure 2.2. Ideally, the smaller the step size the more accurate the derivative calculation, but for complex components requiring large scale simulations, numerical noise hinders the calculation of accurate gradients for small step sizes [24].

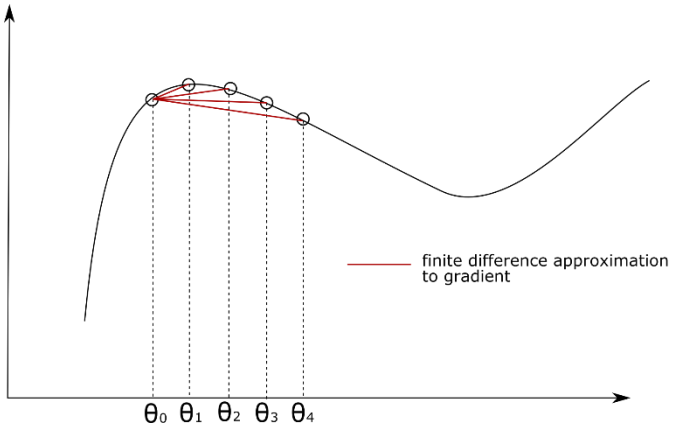


Figure 2.2 Influence of step size on finite differences

2.2 Adjoint methods

In the pursuit of efficient gradient calculations, adjoint based techniques have shown promising results. They have been an area of extensive research over the last two decades, especially for aerodynamic optimization [8-15, 25]. Recently, the applicability of adjoint methods has been demonstrated in turbo-machinery [26-28] and the automotive industry [25, 29-32]. The underlying principle of adjoint methods is the computation of adjoint sensitivities i.e. the derivative of an objective function with respect to design variables. Adjoint surface sensitivity gives information about how the objective function changes for an infinitesimally small movement of each surface mesh node in the normal direction. The development of adjoint methods started with the works of Prof. Pironneau [33] in the field of optimal shape design, and they have followed two different paths (a) continuous adjoint [13, 16, 34, 35], and (b) discrete adjoint [8, 14, 15, 36]. These formulations are based on the process followed to mathematically formulate the adjoint equations from the flow field equations. In the continuous adjoint formulation, the adjoint equations are derived directly from the governing partial differential equations and then discretized, while in the discrete adjoint formulation the governing partial differential equations are discretized first and then the adjoint equations are formulated. In [37] the authors compared the continuous and discrete adjoint approaches for aerodynamic optimization and found that the discrete adjoint gradients are in closer agreement with the gradients computed using finite differences than those computed using continuous adjoint methods, but the difference is small and reduces further as the mesh resolution increases. Also, the authors commented that the computational cost of deriving the discrete adjoint is greater than the continuous adjoint. In the development of continuous adjoint

formulations, Othmer [16] derived the adjoint equations and the boundary conditions for typical cost functions of ducted flows and implemented the results into the open-source finite volume solver OpenFOAM [38] for the computation of adjoint sensitivities.

Full details on the use of the adjoint approach to design are given by Giles and Pierce [8]. An overview of the mathematical formulation of the discrete adjoint approach is presented here. Consider a semi-discrete system of fluid conservation laws described as

$$\frac{d\boldsymbol{U}}{dt} = \boldsymbol{R}(\boldsymbol{U}, \boldsymbol{X}). \tag{2.5}$$

which is referred to as the primal solution. Here $\boldsymbol{X}$ represents the mesh coordinates and $\boldsymbol{U}$ is the vector of the fluid system variables. During the convergence of the primal solution, the non-linear residual $\boldsymbol{R}$ for each equation is driven to zero.

$$\boldsymbol{R}(\boldsymbol{U}, \boldsymbol{X}(\theta)) = 0 \tag{2.6}$$

Differentiating Eqn. 2.6 with respect to design variable θ gives

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\theta} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta} = 0, \tag{2.7}$$

The objective function J depends on the system variables,

$$J = J\big(\boldsymbol{U}, \boldsymbol{X}(\theta)\big). \tag{2.8}$$

The change in performance dJ, due to a change in the value of the design parameter $d\theta$, can be defined as

$$\frac{dJ}{d\theta} = \frac{\partial J}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\theta} + \frac{\partial J}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta}. \tag{2.9}$$

The solution of Eqn. 2.9 using finite differences requires the solution for each design variable. Now, Eqn. 2.7 and Eqn. 2.9 can be combined using an arbitrary vector $\boldsymbol{\psi}$ as

$$\frac{dJ}{d\theta} = \frac{\partial J}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\theta} + \frac{\partial J}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta} + \boldsymbol{\psi}^{\boldsymbol{T}}\left(\frac{\partial \boldsymbol{R}}{\partial U}\frac{d\boldsymbol{U}}{d\theta} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta}\right). \tag{2.10}$$

Eqn. 2.10 can be re-arranged as

$$\frac{dJ}{d\theta} = \frac{\partial J}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta} + \boldsymbol{\psi}^{\boldsymbol{T}}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\theta} + \frac{d\boldsymbol{U}}{\boldsymbol{d}\theta}\left(\frac{\partial J}{\partial \boldsymbol{U}} + \boldsymbol{\psi}^{\boldsymbol{T}}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right) \tag{2.11}$$

The last term of Eqn. 2.11 can be eliminated by choosing a value of $\boldsymbol{\psi}$ such that

$$\frac{\partial J}{\partial U} + \boldsymbol{\psi}^{\boldsymbol{T}}\frac{\partial \boldsymbol{R}}{\partial U} = 0, \tag{2.12}$$

Eqn. 2.12 is referred as the adjoint equation with $\boldsymbol{\psi}$ being the adjoint solution. Now, using Eqn. 2.11 and Eqn. 2.12, the adjoint sensitivities can be obtained as

$$\frac{dJ}{dX} = \frac{\partial J}{\partial \boldsymbol{X}} + \boldsymbol{\psi}^{\boldsymbol{T}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}} \tag{2.13}$$

The important point to be noted here is that the computation of adjoint sensitivities only depends on the objective function (J). Thus, by using adjoint method only one set of additional equations needs to be solved for each objective function, regardless of the number of design parameters. In recent years several adjoint solvers have been developed including adjointFoam [16], SU2 [39], HELYX [40], DLR-TAU [41], HYDRA [42] amongst others.

2.3 Design parameterization

Parametrization is at the core of optimization, as it defines the design space that the optimizing algorithm explores. The success of any shape optimization methodology depends extensively on the type of parameterization technique employed [43]. One straightforward route which results in the most flexible parametrization strategy is to use the nodes of the computational mesh [44-47] as design variables. One major drawback for this parameterization strategy is that, as all surface mesh nodes can move independently, the implementation of a smoothing algorithm is required to prevent the appearance of non-smooth shapes during the optimization process. In this regard, the Free-form deformation (FFD) techniques have been successfully implemented for aerodynamic shape optimization problems [48-50]. These techniques originated from the soft object animation in the computer graphics industry [51]. In this method, a box is created around the object (to be optimized) with a set of control points defined on its surface as shown in Figure 2.3.

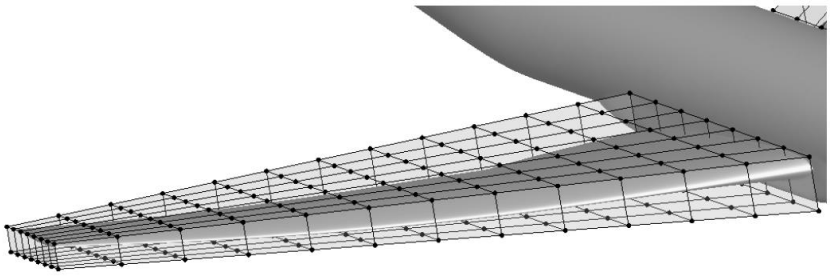


Figure 2.3 FFD box around an aircraft wing (208 control points) [67]

The box is then parameterized as a B*é*zier solid using following expressions:

$$X(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} P_{i,j,k} D_i^l(u) D_j^m(v) D_k^n(w) \tag{2.14}$$

where l, m, n are the degrees of the FFD function, $u, v, w \in [0,1]$ are the parametric coordinates, $P_{i,j,k}$ are the coordinates of the control point (i, j, k), and $D_i^l(u), D_j^m(v), D_k^n(w)$ are the Bernstein polynomials. The benefit of this approach is that it imparts smooth deformations to the analysis mesh and enables the parameterization to alter the thickness, sweep, twist, etc. for the design of an aerospace system. One of the drawbacks of these mesh-based optimization methods is that the mesh topology (in terms of the number of elements present and their connectivity) must remain constant as the model updates. Also, it is the mesh that reaches the optimum shape. This mesh must then be translated into a CAD model before it can be used for further analysis or manufacturing assessments. This mesh-to-CAD step is non-trivial and may require extensive user interaction [52, 53].

In this regard, using CAD geometry within the optimization process should result in a better-quality CAD model shape and aligns with the industrial ambition of having a more integrated design process. Here, the model is always available in the CAD form, and thus do not require any post-processing step of mesh-to-CAD conversion. Some authors [54-58] have attempted to develop optimization processes based on non-uniform rational B-splines (NURBS) patches, where the NURBS control point locations (x, y and z coordinates) are used as design variables (see Figure 2.4).

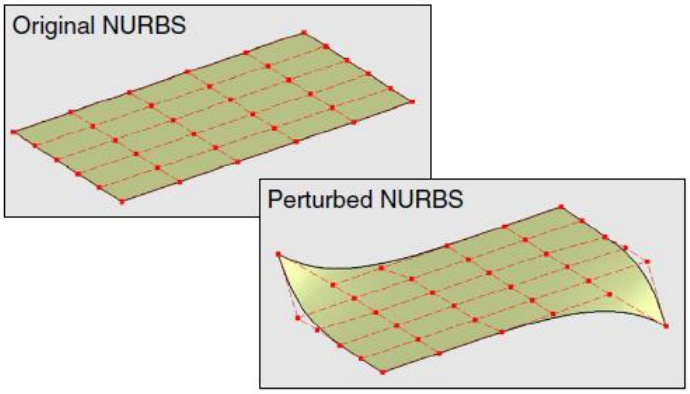


Figure 2.4 A NURBS patch with the net of original (upper left) and perturbed (lower right) control points [56]

NURBS can be defined as

$$X_s(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{i,j} B_{i,j}(u,v) \tag{2.15}$$

where $P_{i,j}$ are the position of NURBS control points, and $B_{i,j}(u,v)$ is the basis function defined as

$$B_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m}N_{k,p}(u)N_{l,q}(v)w_{k,l}}, \tag{2.16}$$

where $N_{i,p}(u)$ and $N_{j,q}(v)$ are the p-th and q-th degree basis functions defined on (u,v) parametric space. In order to increase the applicability of the NURBS for optimization with multiple patches, the approaches have been developed to enforce continuity constraints along the patch interfaces [56]. Recent work by Xu et al. [59] has extended the NURBS parametrisation method to include geometric constraints such as thickness and trailing edge radius. Here the benefit is that the NURBS represent a richer design space than that can be expressed using parameters in a feature-based CAD model. One downside of using NURBS is that sometimes the NURBS control net may be too coarse in certain regions and would require a process to enrich the control net by adding more control points before it is used for optimization.

Jesudasan et al. [60] presented an adaptive parameterization approach based on NURBS patches, where the NURBS control net was refined by using knot insertion, and subsequently used to optimize pressure loss across a U-Bend passage of a turbine blade serpentine cooling passage. In other works, Nurdin et. al. [61] presented an approach to use FFD boxes to directly parameterize the CAD geometry. Koch et al. [62] used NURBS curve to define the level-set boundary and subsequently used it for the shape optimization. Although, the approach demonstrated a link between the 2D level-set topology results and CAD-based shape optimization methods, its extension to 3D models is a challenging task.

The downside of the approaches outlined above is that, as they do not work directly on the parametric CAD model created in a feature-based CAD system. As such the design intent and the parametric associativity captured in the choice of features used to build the model is lost. In the context of CAD modelling, features can be defined in many ways depending on their application [63-65]. Robinson et al. [7] addressed this issue and presented a method to use directly the parameters defining the features in a CAD model feature-tree as design variables. In this approach, the shape of the model was updated by changing the values of the parameters that define it. One of the main advantages of this approach was that the constraints imposed by the features in the

CAD model feature tree will mean that the optimized part can be manufactured. The major drawback of this approach was that it depended extensively on the skills and experience of the CAD model creator, and their ability to visualize and parameterize the design space. Also, the optimized geometry is only a parametric variation of the original CAD features and may require the insertion of additional features into the CAD model feature tree if a radical change in shape or performance is desired.

2.4 Parametric design velocity

Parametric design velocity quantifies the boundary movement with respect to a change in the parameter value. In Figure 2.5, the arrows represent the design velocity as the boundary changes from solid line to the dashed line.

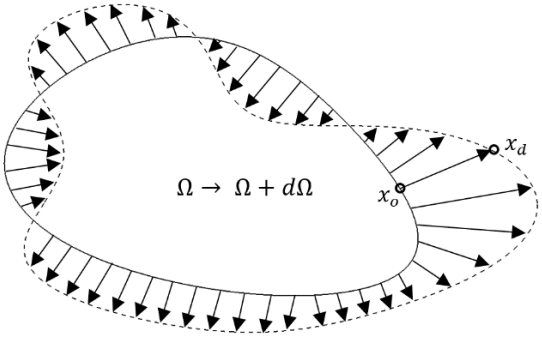


Figure 2.5 A two-dimensional design velocity field

This measure was first developed in the context of structural optimization [66]. Where the motive is to use a parametric CAD model in an optimization framework, the availability of a robust and efficient way of calculating parametric design velocity is of utmost importance. A number of approaches have been proposed in the literature for the computation of design velocity. Chen and Torterelli [66] used an approach based on the parametric position of points on the boundary of the unoptimized CAD model. After a parameter perturbation, the new point position was computed based on the parametric values recorded on the original model. Alternatively, Truong et al. [67] presented an approach for the movement of the surface mesh by comparing the parametric definition of mesh nodes with respect to the tessellation of faces in the original and perturbed CAD model, and subsequently using finite differences to compute the design velocities.

Hardee et al. [68] applied a hybrid of a finite difference method and the boundary displacement method for the computation of design velocities directly from the CAD

model. This involved comparing the parametric description of the faces in the original CAD model with the parametric description after the perturbation of one of the design parameters. One of the problems with the method was that it relied on a one-to-one mapping between the topological entities in the boundary representation of the perturbed and unperturbed geometries, and thus required the boundary topology of the geometric model to remain constant after the model is perturbed. In the context of this thesis, the term "constant topology" refers to the number and arrangement of faces, edges and vertices over the model boundary (or B-Rep) remaining the same. Such a constraint is hard to enforce in practice as the boundary representation is usually computed within the CAD system and is not chosen by the user. An example of a boundary topology change is shown in Figure 2.6, where Figure 2.6(a) shows the unperturbed CAD model and Figure 2.6(b) shows the same model after a parameter has been perturbed. In this example, the boundary topology change is demonstrated by the introduction of the new shaded faces.

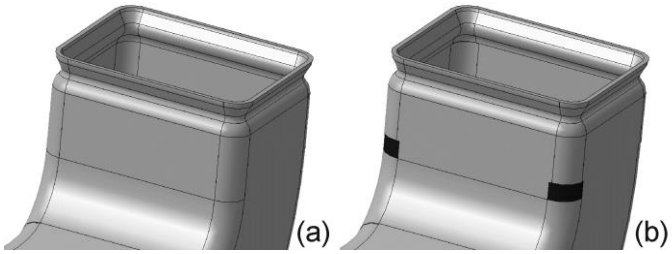


Figure 2.6 Topology change after a parameter perturbation where two new faces are created [7]

Chang et al. [69] computed design velocity using a boundary displacement method for a simplified class of geometric features made up of parametric cubic lines and surfaces. Kripac [70] presented an approach which can be used to compute design velocity from the CAD geometry by assigning certain identities (IDs) to the topological entities in the unperturbed model, and computing velocities from the entities with the same IDs when the model is re-evaluated after a parameter perturbation. This approach is hampered by the persistent naming problem, where the IDs applied to the entities in the CAD model change when it is regenerated after a parameter perturbation. In the cases where IDs are not persistent, techniques are described to rebuild/remap entities based on the construction order of the features in the model using adjacency information. An alternative approach to deal with the persistent naming problem was given by Ragothama et al. [71]. Nemec and Aftosmis [72] who presented an approach for computing the displacement of the boundary based on an embedded boundary

cartesian mesh method, where the movement of the intersections between a cartesian mesh and a mesh of the component geometry is used to determine the boundary movement. This approach was again restricted to problems where the boundary topology remains constant.

Another efficient way to compute design velocities is by directly differentiating the mathematical expressions coded within the CAD systems by using automatic differentiation (AD). AD is a technique to compute analytical derivatives with respect to variables in the computer programs. The main idea of this approach is to analytically differentiate each mathematical operation performed by a computer program and then use the chain rule to automatically accumulate the derivative values. Herein, active variables are declared (variables considered as differentiable quantities) and then the computer program is differentiated to obtain the derivatives with respect to active input variables without any truncation errors. Xu et. al. [56] differentiated an in-house CAD tool based on NURBS patches by using the AD tool Tapenade developed by INRIA Sophia-Antipolis [73]. Sanchez et. al. [74] presented the differentiation of another in-house CAD tool by using AD software tool ADOL-C (Automatic Differentiation by OverLoading in C++) developed at the university of Paderborn [75]. Recently, Banovic et al. [76, 77] demonstrated the differentiation of the open-source CAD system Open-CASCADE (*v7.0*).

These approaches to calculate design velocity have significant advantages as they do not require a geometry or mesh to be recomputed. They are both efficient and robust against boundary topology changes. These approaches also avoid the difficulties with numerical accuracy that are associated with finite difference approaches. The downside is that they require access to the underlying source code of the CAD system, which is unlikely to become an industrial reality for the major CAD systems in the near future. Further, the derivative calculation for complex geometric configurations and Boolean operations is still a challenging task.

Another approach for design velocity computation was presented by Robinson et al. [7], where discrete representations of CAD model boundary were used. In this approach, geometric facets in the Virtual Reality Modelling Language (VRML) format were exported directly from the CAD system, and the design velocity was computed by comparing the unperturbed and perturbed models. The design velocity was calculated at the nodes of the facetted model and then associated back to the CAD

geometry. The advantage of this approach was that it was able to overcome the restrictions associated with the boundary topology changes and with the persistent naming problem. But, the key limitation was that the approach was not able to calculate design velocity for shape changes which the initial faceting was unable to represent. This can occur when the parameterization allows a face to curve at a much greater resolution than the faceting of the model represented. Also, the generated facets were too large in size resulting in non-smooth design velocities. For example, in Figure 2.7(a) the top face of the block is initially flat and modelled by two facets with nodes at the corners. These facets are unable to capture the subsequent curvature of the perturbed face shown in Figure 2.7(b).

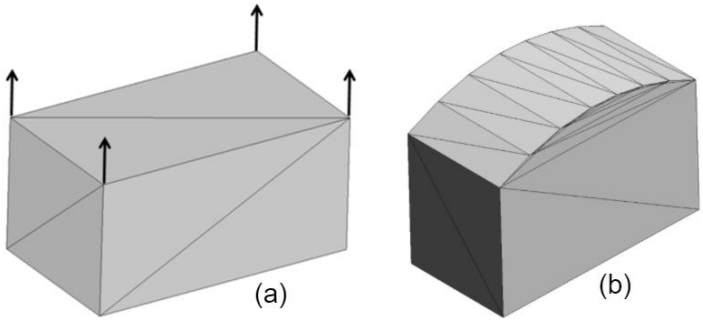


Figure 2.7 (a) Top surface represented by two facets with all nodes at surface corners (b) Modified shape of the top surface not captured by the faceting (design velocity is zero at all nodes)

Thompson [78] in his PhD research addressed these issues and made advances by using tri-surface meshes (or facets) instead of VRML facets for representing the model when computing design velocity. Thompson exported CAD models in STEP [79] format from the CAD modeler which were then tri-meshed with a commercial mesh generating software CADfix [80] to generate triangular surface facets. The STEP format is a CAD translation standard that does not include any features or parameters. The facets produced using the method were more uniform compared to that produced using VRML format, which enhanced the capabilities to capture small geometrical changes. Also, Thompson calculated design velocity at the centroid of smaller and uniform facets compared to the nodes of coarse facets as used in [7]. Though the developed approach was efficient in many ways, the result strongly depended on the values of parameters used for generating facets (sag, turn and length) which needed to be set to different values depending on the CAD model being optimized. Also, the approach was unable to produce design velocities when the perturbation caused the

model to update such that the projection of a point on original model in the normal direction lies outside the perturbed model as shown in Figure 2.8, where the symbol "Δ" represents a region of the boundary for which there is no obvious projection after the perturbation shown (from solid to dashed). During the initial phase of this thesis, developments are described (presented in chapter 3) which enhance the previous approach for calculating design velocity and addressed the aforementioned limitations to increase its robustness. Also, a link was established between the developed tool with an open-source surface mesh generator to facilitate its use by other researchers.

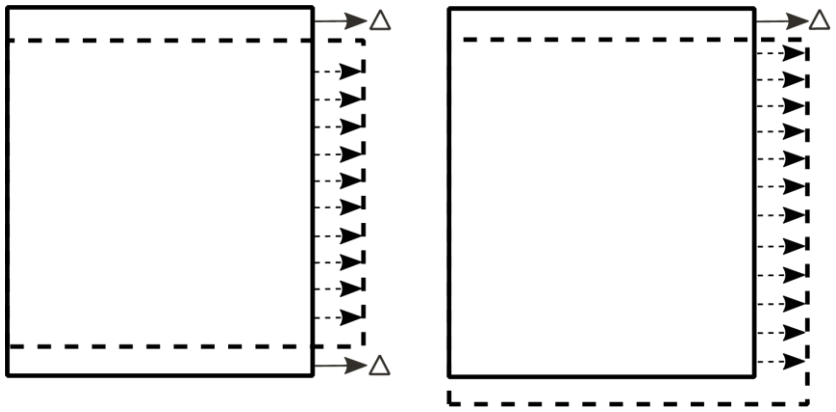


Figure 2.8 Geometrical movement when the design velocity fails: original (solid line) & perturbed model (dashed line)

2.5 CAD feature modelling

In a feature-based CAD modelling system, a part model is comprised of individual features which are combined to represent an overall shape. A wide range of CAD features can be defined such as pads, pockets, holes, fillets, chamfers etc. These are mainly classified as sketch-based features or dress-up features. Sketch-based features are created by defining a 2D sketch profile and a 3D model is generated from the sketch using extrusion, rotation, sweeping or lofting. Dress-up features, like fillets and chamfers, are created directly on the solid model.

When creating a model within a CAD modelling system, it is common to create relationships to specify that the value of one parameter is a function of the values of other parameters in the model. This relationship is part of the design intent of the model and once applied the parameters cannot be controlled independently. In the process of model generation, the CAD system automatically creates a series of parameters and relationships in the background. The number of such parameters can range from hundreds to thousands depending on the model complexity. Most of these parameters

have no influence on the shape or size of the model and are just the representative quantities. The parameters which are responsible for changing the model's shape are defined with real, integer and Boolean values. These parameters can be assessed by using a suitable CAD system application programming interface (API). In this thesis, only continuous dimensional parameters which affect the shape of the model are considered. Modifying parameters with real values (such as lengths and angles) by small amounts will typically cause movements of the boundary proportional to, and of the same order as, the size of the parameter perturbation.

In this thesis, the choice of CAD system is motivated by the involvement of industrial partners Rolls-Royce Deutschland (RRD) and Volkswagen Group Research (VW), where SIEMENS NX and CATIA V5 are used as the CAD modelling system. Here, a discussion is presented about these CAD systems with emphasis on their capabilities to create a feature-based CAD model.

2.5.1 CATIA V5

CATIA (Computer Aided Three-dimensional Interactive Application) V5 R21 developed by Dassault Systems is a 3D CAD modelling system. It provides a platform for collaborative product creation and product data management. It is also commonly known as 3D Product Lifecycle Management software as it supports multiple stages of product development starting from an initial concept to the final manufactured product. It is extensively used in a variety of industries including aerospace, automotive, consumer goods, and industrial machinery.

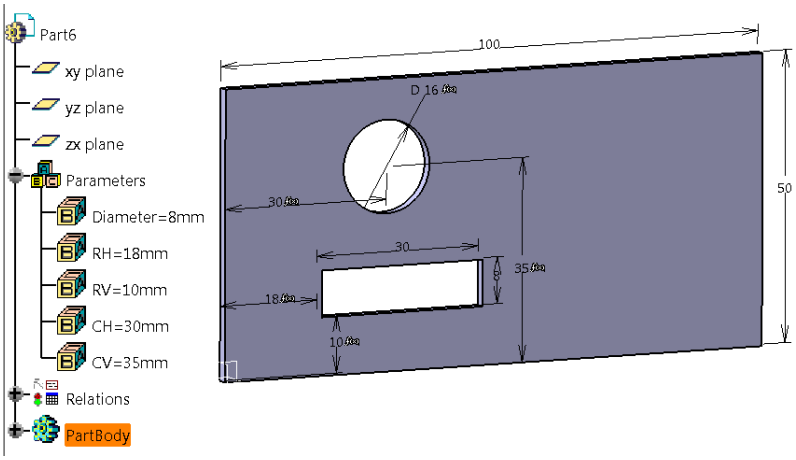


Figure 2.9 CATIA V5 feature tree representation

In CATIA V5 the CAD features are stored in a feature tree as shown in Figure 2.9, and a list of all of the parameters used to construct the model can be found through the

"Formulas" window as seen in Figure 2.10. It should be noted that for the model in Figure 2.9, all of the model feature parameters (e.g. diameter of the hole, length and depth of the slot) are belonging to the sketch.

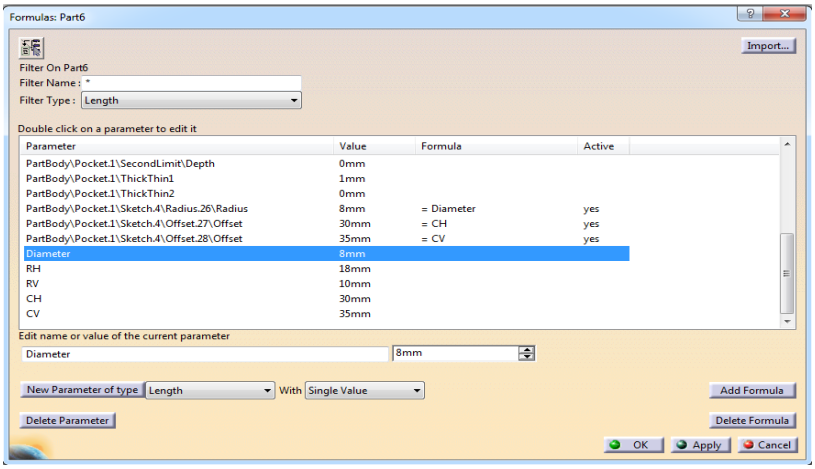


Figure 2.10 CATIA V5 parameters definition

2.5.2 Siemens NX

Siemens NX is a CAE software package for design, simulation, and manufacturing solutions. NX allows the parametric and feature-based modelling capabilities for complex mechanical designs. It is extensively used in various industries including aerospace and defence, automotive, electronics, marine, medical devices etc. In Siemens NX the features used to create the model are found in "Model History", and parameters defining these features are found in "User Expressions" as shown in Figure 2.11.

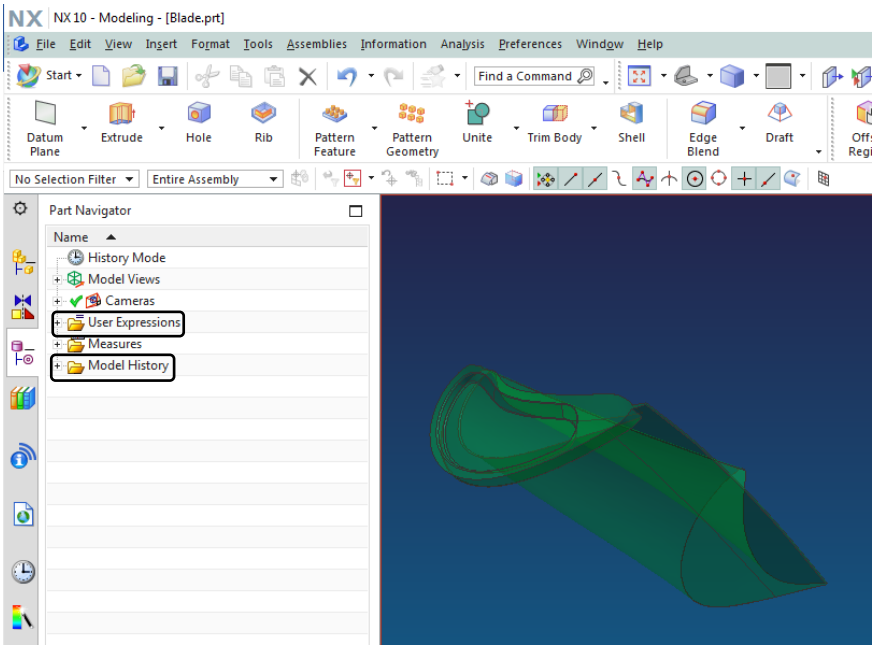


Figure 2.11 CAD model in SIEMENS NX

2.6 Software Used

2.6.1 HELYX

HELYX is a comprehensive general purpose CFD software package based on ENGYS open-source CFD simulation engine. HELYX-adjoint is a continuous adjoint CFD solver which delivers both surface and volume sensitivities for pre-defined objective functions and flow constraints including: minimisation of power losses, maximisation of flow uniformity, minimisation of drag force, maximisation of turbomachinery efficiency, maximisation of flow rate, equalisation of flow split across multiple outlets, etc. It can be used for both topology and shape optimization and is based on OpenFoam technology. In this thesis, HELYX is used for the analysis of test-cases from IODA industrial partner VW.

2.6.2 HYDRA

HYDRA is a Rolls-Royce in-house CFD solver based on discrete adjoint formulation. It is a nonlinear flow solver using a node-based finite-volume discretisation method and a pseudo-time-marching scheme to reach steady state, accelerated by a block-Jacobi preconditioner and a geometric multigrid technique. HYDRA has been successfully applied on industrial test cases [81, 82], and more details on the underlying theory and implementations can be found in [9, 83]. In this thesis, HYDRA is used for analysis of the industrial test-cases from RRD. Please note that the primal and adjoint CFD analysis was done at RRD and only the adjoint sensitivities were provided to QUB.

2.6.3 Stanford University Unstructured (SU2)

The SU2 code is an open source CFD analysis tool for aerodynamic shape optimization. SU2 uses a finite volume method for the spatial discretization of partial differential equations, with a standard edge-based structure on a dual grid. The convective and viscous fluxes are evaluated at the midpoint of each edge in the mesh and then integrated to evaluate the residual at every node in the mesh. The SU2 suite is also able to solve the continuous adjoint Euler/RANS equations. The adjoint solver can produce surface sensitivities for a range of objective function including drag, lift, side force, efficiency, moment etc. In this thesis, SU2 is used as the state-of-the-art CFD solver used for the analysis of aerodynamic flows.

2.6.4 ABAQUS CAE

ABAQUS CAE [84] is a finite element modelling and visualization package distributed by DASSAULT SYSTEMS. It provides an interface both to create geometry and to import CAD models for meshing or integrate geometry-based meshes that do not have associated CAD geometry. It provides an interface for CAD models created in CATIA V5, SolidWorks, Pro/ENGINEER etc. and for neutral CAD formats like STEP, IGES, Parasolid etc. ABAQUS CAE also offers comprehensive visualization options, which enable users to interpret and communicate the results of any Abaqus analysis. In Abaqus, python is used as the scripting language to enable automation. In this work, ABAQUS CAE is used to solve several structural mechanics problems with strain energy density as the objective function.

2.6.5 Computational environment

In this work, the Scientific Python Development Environment (Spyder) with Python 3.5 is used as the computational environment. Spyder is a powerful interactive development environment (IDE) for the Python language with support of interactive Python (IPython) interpreter. The numerical analysis was performed using NumPy and SciPy Python libraries.

2.7 Research methodology

Based on the literature reviews, it is evident that the adjoint methods play an important role in the development of optimization methodologies, but their integration into the regular development processes is yet to be accomplished. The IODA project aimed to develop methodologies dealing with the parameterization and geometrical modelling aspects that are necessary to integrate the adjoint solvers in the industrial design workflows. The research presented in this thesis is part of the IODA work package (WP) 4, 5 and 6 dealing with CAD-based parameterization, constrained optimization and industrial workflows.

In an industrial workflow, the part design starts from a CAD geometry and has to eventually deliver the optimized geometry in CAD. There are some key advantages of having a CAD model as the output of the optimisation process: (i) it can be more easily linked to the remainder of the design enterprise, and (ii) the CAD packages have specialist functionality to be able to assess certain aspects of the optimized design. But,

there are a few challenges associated with the use CAD models such as how to compute the design velocities to measure the movement of the model boundary in response to the perturbation of CAD parameter. Also, from the literatures it is evident that the parameterization is vital for the success of any optimization framework. Where the motive is to directly use the parametric CAD models for the optimization, it is important to develop methodologies that can be used to automatically enrich the design parameterization and if needed automatically insert new features into the CAD model feature tree. This thesis contributes towards the enhancement of the state-of-the-art methods in the CAD-based adjoint optimization methods, formulating a research methodology as outlined in Figure 2.12.

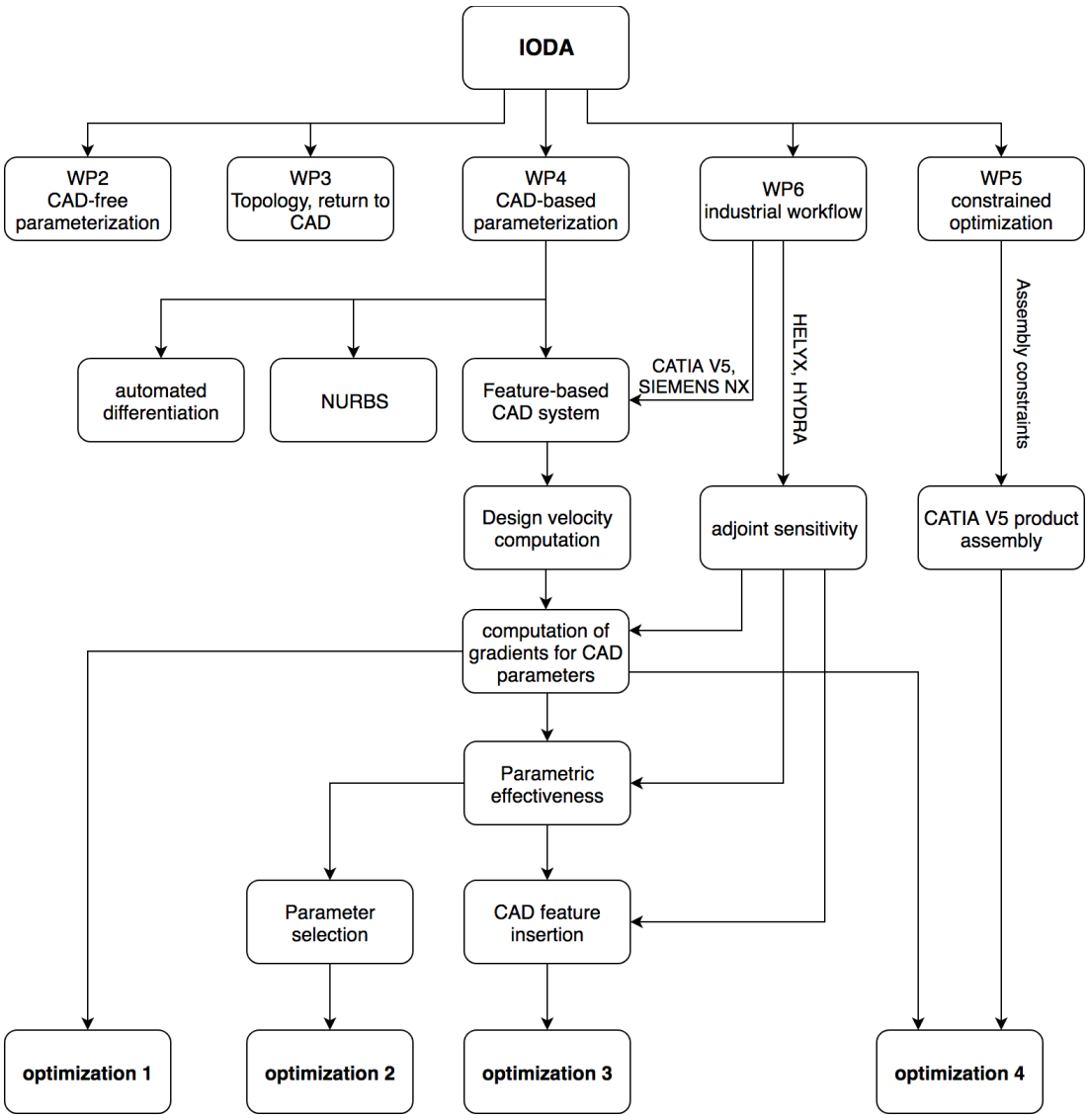


Figure 2.12 Research methodology

2.8 Thesis aims and objectives

The research herein aims to contribute to the overall aim of IODA by developing efficient methodologies to facilitate the use of commercial CAD systems within the optimization framework. The objectives of this research are summarized as:

1. Develop a robust and efficient methodology to calculate design velocities for parameters defined in a CAD model feature tree for complex components and link them to the adjoint sensitivities to use in a gradient-based optimization framework.
2. Develop methodologies to automatically rate the quality of CAD parameterization and select the optimum combination of CAD parameters to be used for optimization.
3. Develop methodologies to automatically add the optimum new CAD features to the model in order to improve the manner in which the shape can update.
4. Develop methodologies to consider the constraints imposed on a design from adjacent components in the product assembly.

2.9 Summary

This chapter has presented a review of literature in the areas of adjoint methods, highlighting their prospects to be used in an industrial design workflow. Moreover, existing methods for design parameterization and computing design velocity were reviewed. In past years, there has been interest in using CAD models for optimization, but their applicability has been restricted owing to several reasons. One of them is the lack of methods for optimizing directly on CAD geometries. The current methods using CAD models for optimization are not robust enough to be employed in an industrial workflow, where it is common to use commercial CAD software. Further, the current methods for generating a CAD model from the optimization performed on CAE meshes are less efficient and lose important geometric details.

In the following chapters, new approaches are described which facilitate the use of commercial CAD software within an optimization workflow. An initial investigation is done to overcome the limitations of existing approaches and develop a robust and efficient method for computing design velocity for CAD model parameters. This is then used to present a novel methodology to select the optimum set of CAD parameters

to be used for optimization. Thereafter, novel methodologies are outlined to automatically increase the design flexibility of CAD models by inserting optimum CAD features directly into the CAD model feature tree. Further, an optimization framework is presented which incorporates constraints imposed by the adjacent components in the CAD product assembly, which the component being optimized is not allowed to violate. This is important in industrial workflows, where a mechanical interference detected during the product assembly, can lead to excessive re-work and may also reduce the performance of the component.

Chapter 3 Design velocity and gradient computation

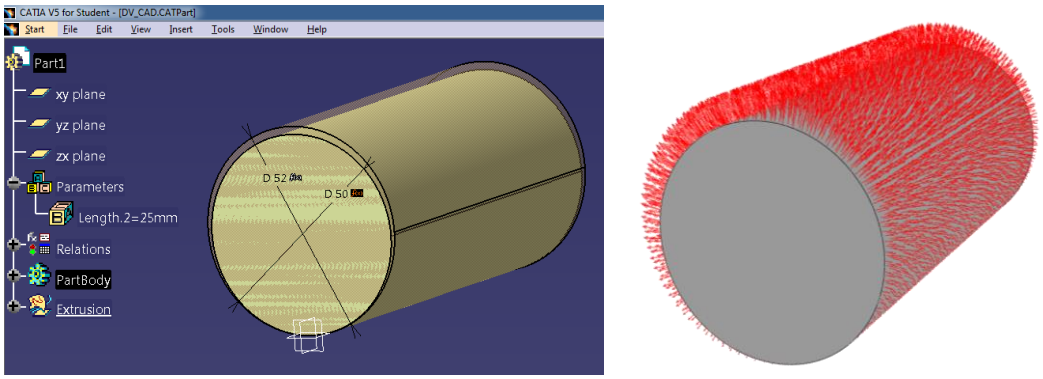
3.1 Introduction

This chapter concentrates on the development of a robust methodology for the calculation of design velocities for the parameters defined within a commercial feature-based CAD modelling system. The design velocities are then linked with the adjoint surface sensitivities to obtain the performance gradients of the parameters required for the optimization.

In the context of this thesis, design velocity V_n represents the normal component of the boundary movement calculated as

$$V_n = \delta \boldsymbol{X}_s \cdot \hat{\boldsymbol{n}}, \tag{3.1}$$

where $\delta \boldsymbol{X}_s$ is the boundary movement and $\hat{\boldsymbol{n}}$ defines the direction of surface normal at that point. For each location on the domain boundary, the design velocity is represented by a scalar value.



(a) (b)

Figure 3.1 Parametric CAD model, (b) vector representation of design velocity

Figure 3.1(a) shows CAD model of a cylinder in solid yellow, where the location of the bottom of the defining sketch is defined to be at the origin. The transparent shape superimposed is the model after the radius defining the cylinder is changed from 25mm to 26mm. In Figure 3.1(b), the arrows represent the design velocities as the

boundary changes from the original to the perturbed model. The convention adopted throughout this thesis is that a positive design velocity represents an outward movement of the boundary, and negative is inward.

It should be noted that the parametric design velocities are only consistent if the CAD model is fully defined. When the model is under-defined the effect of a parameter perturbation may not be unique and the result may differ depending on the CAD system being used or the magnitude of the perturbation. For example, Figure 3.2(a) shows an under-defined model with one variable parameter *d*. Its edges are horizontal or vertical and no other constraints are applied (i.e. it is not fully defined). Figure 3.2(b) and Figure 3.2(c) show two different results which may be possible when parameter *d* is perturbed by *p*.

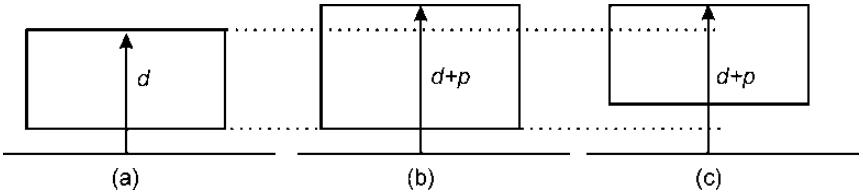


Figure 3.2 Perturbing under-defined models

3.2 Design velocity computation

As it is common for industrial CAD models to be defined by hundreds (or even thousands) of parameters, a computationally efficient approach (which can be run in parallel to the CFD analysis) for calculating design velocity is required if it is to be used for optimization. Herein, the design velocity is calculated using a finite difference approach based on the CAD model before and after a parameter perturbation. The CAD geometries are represented using a surface tessellation of linear triangular elements referred as faceting. A faceted representation of CAD geometry is created by employing an open-source meshing tool GMSH [85]. The Open Cascade geometry kernel in GMSH enables the use of CAD geometries in the following format: IGES, STEP, B-Rep, and its own GEO format. The mesh generation in GMSH is performed in the bottom-up manner i.e. the nodes are positioned at the vertex first, then the lines are discretized, followed by the surface mesh and then the volume mesh.

The displacement of the model due to a parameter perturbation is approximated by calculating how much a point at the centre of each facet in the unperturbed model must move to reside on the boundary of the perturbed model. The key requirement is that the faceting of the unperturbed model should be of sufficient resolution to capture the

curvatures in the original model, and in each of the perturbations of the model. The first is important as the normal direction over the boundary of the unperturbed model in Eqn. 3.1 is calculated from the facets. The inability to capture the curvatures in the perturbed models hampered the approach in [7] as such sizing information is difficult to gauge a priori. In this work, two different approaches are employed for defining the surface facet distributions: (1) a fixed target element size, (2) created using density boxes ensuring high mesh density in the areas of high curvature. Figure 3.3(b, c) shows the surface facets created using the two approaches on the ONERA M6 CAD model shown in Figure 3.3(a).

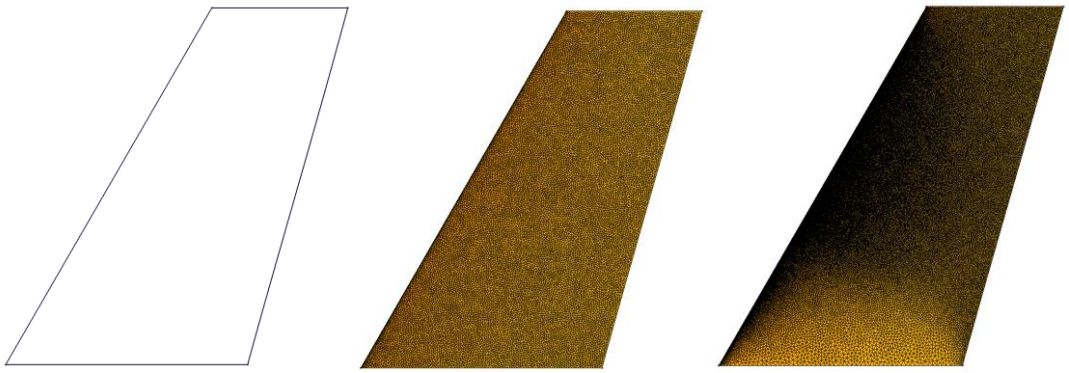


Figure 3.3 ONERA M6 (a) CAD model, (b) coarse facets, and (c) fine facets

To incorporate various CAD modelling systems, a generic representation of the CAD model (the STEP format) is used as the input to the calculation of design velocities. Using a CAD modelling system, STEP files are created for each parametric perturbation to be used for the calculation of design velocities. The perturbation of a model feature parameter and the export of the corresponding STEP file is automated using a CAD system API. This approach requires the CAD parameters to be perturbed to change the shape of the geometry, but sometimes these changes can be highly non-linear (particularly for angular parameters), making the approach sensitive to the step-size used for perturbation. In this thesis, in each case the perturbation size is selected to be small ($0.1\% - 1\%$) relative to the size of the features in the model, and different step sizes are experimented with to find the one which gave consistent results i.e. similar design velocities on reducing the step-size. The algorithm for calculating the design velocity from the resulting STEP files is described in Algorithm 1 and is implemented in Python 3.5.

Note: In the processes described below 3.2.1, 3.2.2, 3.2.3 and 3.2.5 have been reimplemented based on the works of Thompson [78], and 3.2.4 is a new technical contribution of this work.

Algorithm 1: Design velocity Computation

Input: Parametric CAD model
Output: Design velocity for each CAD parameter

```
Python CAD API interacts with CAD modeller to extract parametric information of the model (n parameters).
Generate STEP file for original model.
Perturb each parameter by a small value (±ϵ) and generate STEP files (n).
Generate surface facets for each STEP files (n + 1) using GMSH.
Read surface facets for original geometry.
Compute centroid and normal vector for each surface facet (N facets)
for parameter i ← 1 to n do
    Read surface facet for i^th geometry
    Compute centroid and normal vector for each surface facet
    generate KD Tree for each centroid
    for facet j ← 1 to N in original model do
        while projection unsuccessful do
            project j^th facet centroid in the normal direction onto facet of i^th geometry
            if projection is successful then
                Design velocity calculated
            else
                select next facet candidate
```

3.2.1 The projection tests

The displacement of the model due to a parametric perturbation is calculated by projecting a point at the centroid of each facet in the unperturbed model in the normal direction onto the facet in the perturbed model.

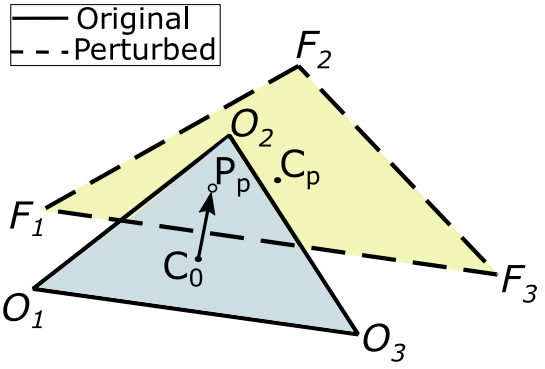


Figure 3.4 Projection from unperturbed facet centroid C_0 to perturbed facet with centroid C_p to get the projection point P_p

In Figure 3.4, a facet on an unperturbed model defined as $\Delta O_1 O_2 O_3$ has its centroid C_0 projected onto the facet $\Delta F_1 F_2 F_3$ in the perturbed model to find a projection point P_p. The coordinates of P_p are computed using

$$P_p = C_0 + \left\{ \frac{(C_p - C_0).\hat{n}_{C_p}}{\hat{n}_{C_p}.\hat{n}_{C_0}} \right\} \hat{n}_{C_0}. \tag{3.2}$$

To determine if a given point P_p lies inside the perturbed facet, consider the Barycentric coordinates [86] as shown in Figure 3.5. The Barycentric coordinates ζ, η and ξ are computed using the procedure described by Ericson [86], and the values of these parameters represent different regions. For example, ζ varies from 0 to 1 along F_1F_3, η varies from 0 to 1 along F_1F_2 and ξ varies from 0 to 1 along F_2F_3. The point P_p lies inside the perturbed model facet if $\{\zeta > 0;\ \eta > 0;\ \zeta + \eta < 1\}$. If a projection is found to be contained within the boundaries of a triangular facet, then the normal vectors of the facet in the unperturbed and perturbed models are compared to determine if they lie within a small angular threshold. If both the conditions are satisfied (i.e. the projected point lies within the triangular facet and the surface normal is within the specified tolerance) the projection is deemed to be successful. If not, a search is then conducted on an adjacent perturbed facet using the same criteria.

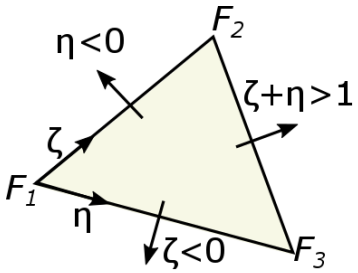


Figure 3.5 Using Barycentric coordinates to determine which facet to test next

3.2.2 Determining which facet in the perturbed model to test first

One of the goals of this approach is to overcome two main limitations that restricted the applicability of some of the approaches mentioned in section 2.4. These are (1) the need for the model's boundary topology to remain the same before and after a parameter perturbation; (2) the need for the facet labels and their correspondence to the model geometry to guide the projection. This is achieved by projecting the unperturbed facet centroid onto the perturbed facets after each parameter change. To determine which facet on the perturbed model to use for the projection requires a search operation over the discretized facets (as opposed to relying on face labels or boundary topology). This is achieved by setting up a multi-dimensional binary search tree (KD-tree [87]). For each perturbed model, a KD-tree of its centroid point coordinates is created. A KD-tree query returns for each facet centroid in the

unperturbed model, the closest facet centroid in the perturbed model. The first projection test is performed using this facet. If the projection test is successful, the facet label and the coordinates of the projection point is recorded.

3.2.3 Determining which facet in the perturbed model to test next

If the projection is unsuccessful for the selected perturbed facet, the Barycentric coordinates are used to determine which facet to use for the next projection test. With reference to Figure 3.5, the next facet to test is selected according to

- $\zeta < 0$ the adjacent facet which shares the vertices F_1 and F_3 should be tested next,
- $\eta < 0$ the adjacent facet which shares the vertices F_1 and F_2 should be tested next,
- $\zeta + \eta > 1$ the adjacent facet which shares the vertices F_2 and F_3 should be tested next.

This sequence of projections and identification of facets to test next continues until the projection is successful. This step can continue till all the facets in the perturbed model are tested. In order to be computationally efficient, this step is terminated if the number of unsuccessful projections reaches a threshold value before a brute force approach is employed. The brute-force approach requires the centroid point to be projected onto all the facets in the perturbed model and selecting the closest facet which it projects onto with the similar facet normal. In order to limit the number of facets tested, only some of the facets in the perturbed are tested, and the threshold value has been set to 500 facets for the test cases in this thesis.

3.2.4 Design velocities when projections fail

In some cases, it is possible that even the brute force approach is not able to yield a successful projection. This is a possibility in cases where the perturbation causes the model to update such that the direction of the normal vector at a point on the original model points outside the perturbed model, as depicted in Figure 2.8. In these cases, a small number of nearby facets in the perturbed model are tested (in this work fixed to 200), and if their surface normal is found to lie within a prescribed tolerance (approximately 2°), the un-perturbed facet centroid is projected onto the facet the centroid belongs to and the design velocity in the normal direction is calculated

accordingly. A final condition is introduced if facets of the unperturbed model are still unable to be projected onto the perturbed model. In such cases the design velocity is interpolated from those of the neighbouring facets in the unperturbed model which share the common vertex with the original facet.

3.2.5 Computing design velocity

Once the unperturbed facet centroid (C_0) has been successfully projected in the normal direction ($\hat{\boldsymbol{n}}_{C_0}$) to obtain the projection point (P_p) on the perturbed model, the design velocity at C_0 is calculated using Eqn. 3.1, which results in

$$V_{n,0} = \left(P_p - C_0\right) \cdot \hat{\boldsymbol{n}}_{C_0} \tag{3.3}$$

An automated workflow was developed to compute the parametric design velocities for the parameters defining the CAD model as shown in Figure 3.6.

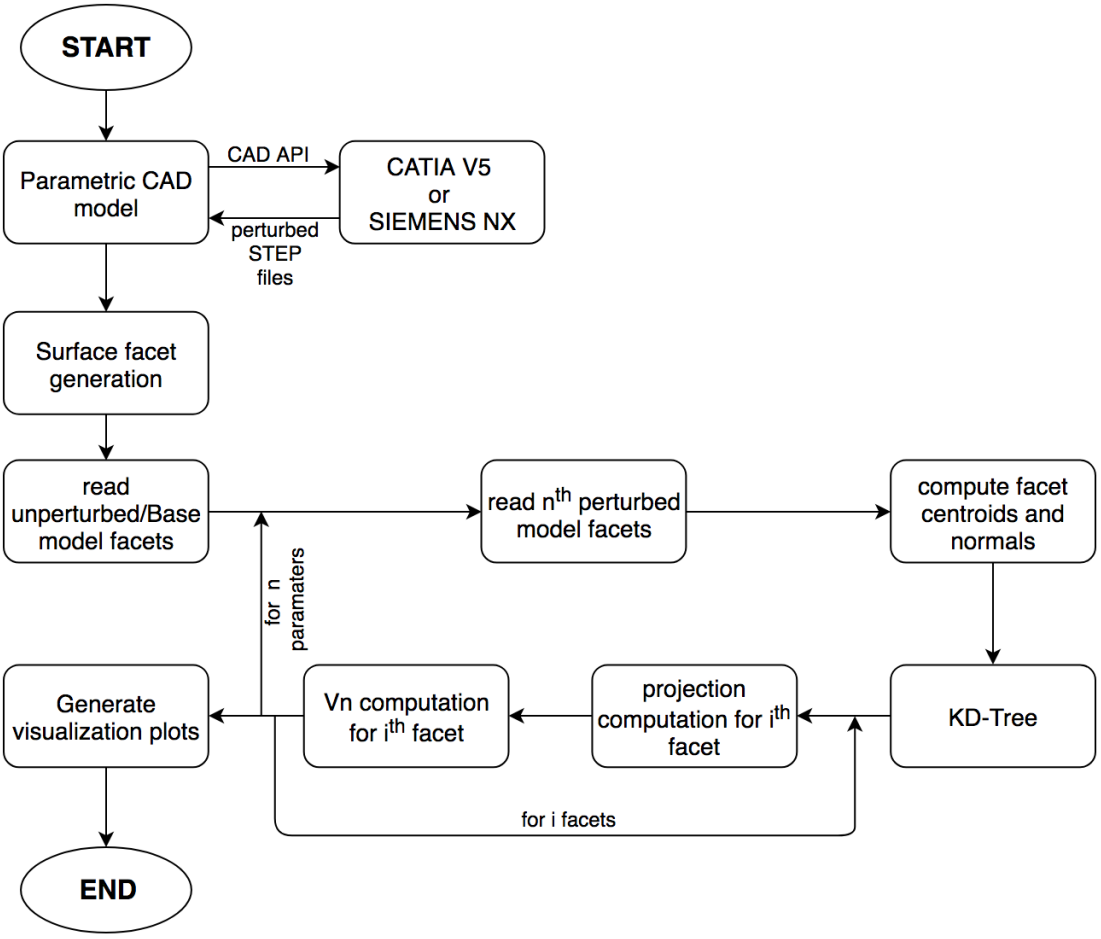


Figure 3.6 Flow chart for design velocity computation

It should be noted that the presented methodology uses only the information of the facets (centroids and vertices) in the perturbed model and neither uses the face or facet labels, nor the correspondence of the CAD model topology between the perturbed and

unperturbed model to compute the design velocity. Hence, making this strategy more efficient and robust than the current alternatives discussed in Chapter 2.

3.2.6 Advancement over existing design velocity approach

As discussed in section 2.4, the original approach for the design velocity computations [78] was unable to produce the correct design velocities for the cases where the geometry is either compressed or perturbation causes the model to update in a way that the normal at a point on original model lies outside the perturbed model as depicted. In Figure 3.7, a simple example of a rectangular box is shown where the initial model is depicted in solid lines and perturbed model in dashed line. The resulting design velocities can be seen in Figure 3.8 where red signifies movement in the normal direction, blue signifies movement opposite to normal direction while green signifies zero design velocity. The original technique fails in these situations as some of the mesh elements do not find a successful projection along the normal direction and hence possess zero design velocity (green strip on the top face instead of red).

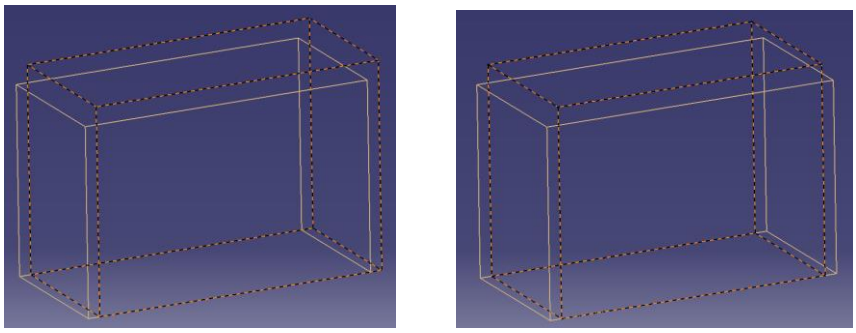


Figure 3.7 Geometrical movement when DV fails: original (solid line) & perturbed model (dashed line)

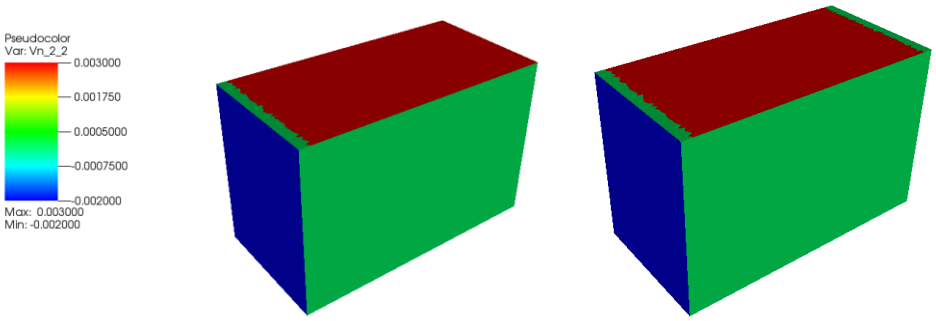


Figure 3.8 Failed Design Velocity predictions

The overall process of calculating design velocities is depicted in Figure 3.6 and results with the modified code are shown in Figure 3.9. The green strips are no longer present on the top face and design velocity is continuous as expected.

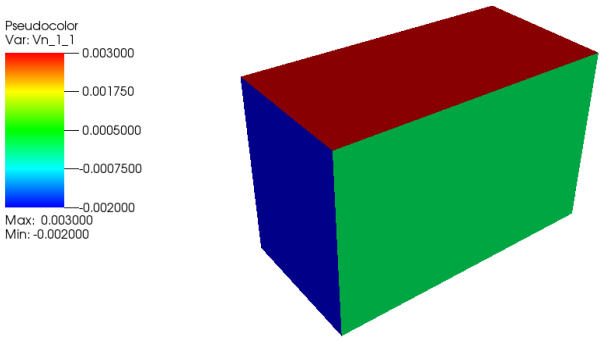


Figure 3.9 Design velocity predictions with modified code

3.3 Validation of design velocity

To validate the accuracy of the computed design velocities, the methodology described in the preceding section is compared against the analytical results for a number of test cases.

3.3.1 Test Case 1

Test case 1 is a plate created by extruding a profile created using three straight lines (bottom and sides) and a Bézier curve along the top. The Bézier curve is defined using five control points, evenly distributed along the top edge as shown in Figure 3.10.

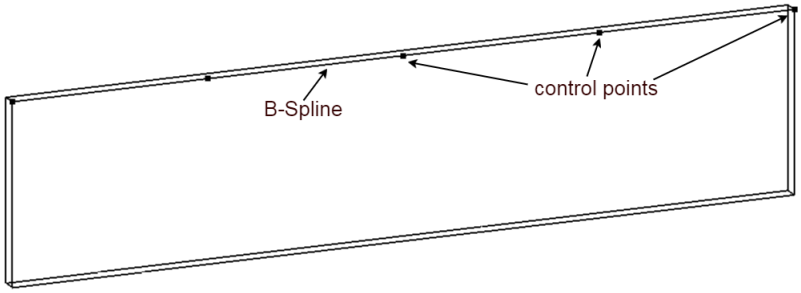


Figure 3.10 Plate model with Bézier control points

The purpose of this simple parameterization is that it allows the comparison with the analytical design velocities computed using the mathematical description of a Bézier curve as

$$P(\varsigma) = \sum_{i=0}^{n} \beta_i D_{in}(\varsigma), \tag{3.4}$$

where P is the point on the curve, β_i is the i^{th} control point, and $D_{in}(\varsigma)$ is the Bernstein polynomial of degree n and $\varsigma \in [0,1]$. Using Eqn. 3.3, the z-displacement can be calculated at each surface mesh node for a perturbation of the control points. A series of design velocity fields are calculated for the CAD model by perturbing each control

point in the z-direction. A vector plot of the design velocity field corresponding to the movement of a control point at the centre of the curve is shown in Figure 3.11. The design velocity field computed using the approach outlined in this chapter, is compared with the analytical results as shown in Figure 3.12 for the perturbation of two of the control points. The two approaches give results with a maximum error in the order of $10^{-7}m$ for a perturbation of $10^{-3}m$.

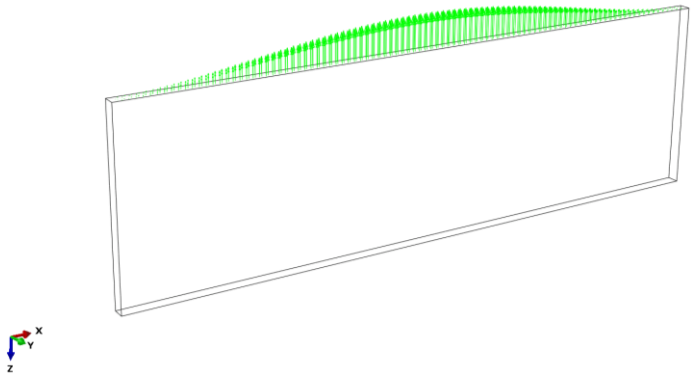


Figure 3.11 Design velocity vectors for parameter perturbation of +1mm

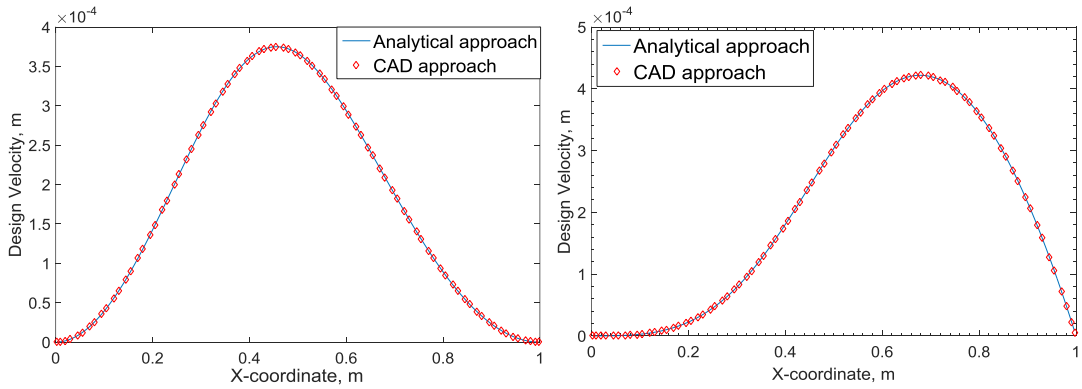


Figure 3.12 Comparison between analytical and CAD based design velocity (CAD results plotted for every other point): (a) X=0.5; (b) X=0.75

3.3.2 Test Case 2

The applicability of the developed methodologies is further demonstrated for a constant sweep aerofoil with twist as the design parameter. A CAD model of a 3D wing is constructed by extruding an aerofoil section defined by two B*é*zier curves (one for the upper surface and one for the lower surface) as shown in Figure 3.13.

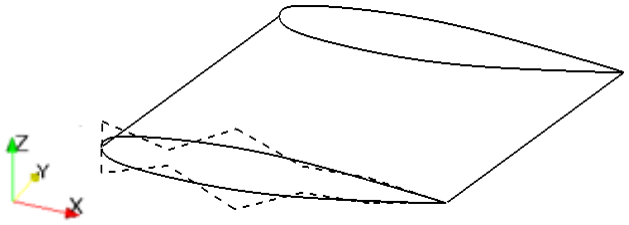


Figure 3.13 CAD model of wing with B*é*zier control points

The boundary of a 2D section along the wing span is described as a B*é*zier. The tangential direction at a point on the curve can be calculated by differentiating Eqn. 3.4 with respect to x, y and z. The normal vector at the point can be obtained by rotating the normalized tangential vectors as

$$\hat{N}_x = d\hat{P}_x \cdot \cos\frac{\pi}{2} - d\hat{P}_z \cdot \sin\frac{\pi}{2} = -d\hat{P}_z(\varsigma), \tag{3.5}$$

$$\hat{N}_z = d\hat{P}_x \cdot \sin\frac{\pi}{2} + d\hat{P}_z \cdot \cos\frac{\pi}{2} = d\hat{P}_x(\varsigma). \tag{3.6}$$

Considering the twist is controlled by the rotation of the wing tip about the Y-axis, which is located along the leading edge, as the design parameter by φ_0. The position of a point (x_i, y_i, z_i) on the wing can be defined in terms of twist φ_i by using the coordinate transformation matrix to obtain the point position (X_r, Y_r, Z_r) as

$$\begin{Bmatrix} X_r \\ Y_r \\ Z_r \end{Bmatrix} = \begin{Bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{Bmatrix} \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}. \tag{3.7}$$

φ varies linearly along the wing span according to

$$\varphi_i = \frac{y_i - y_{root}}{y_{tip} - y_{root}} \varphi_0. \tag{3.8}$$

The derivative of the point position on the wing body relative to φ can be calculated as

$$\begin{gathered} dX = (-x_i * sin\varphi_i + z_i * cos\varphi_i) * d\varphi \\ dY = 0 \\ dZ = (-x_i * cos\varphi_i - z_i * sin\varphi_i) * d\varphi \end{gathered} \tag{3.9}$$

The design velocity can then be computed using Eqns. 3.6, 3.7 and 3.10 as

$$V_n = dX \cdot \hat{N}_x + dZ \cdot \hat{N}_z. \tag{3.10}$$

The movement of CAD geometry caused by the wing twist is shown in Figure 3.14, where solid and broken lines represent original and perturbed geometries respectively. For the design velocity calculation, a perturbation value of 0.5° is used to twist the wing. The surface tessellations are created in GMSH using background mesh with refinement boxes resulting in a total of 126,289 facets. The difference between the design velocities computed using the approach described in this chapter and those computed analytically for the wing twist are shown in Figure 3.15. The two approaches give results with a maximum difference in the order of 7×10^{-5} m. The maximum displacement of the boundary caused by this parametric perturbation is in the order of 10^{-3} m for a wing of span 1m and chord 1m.

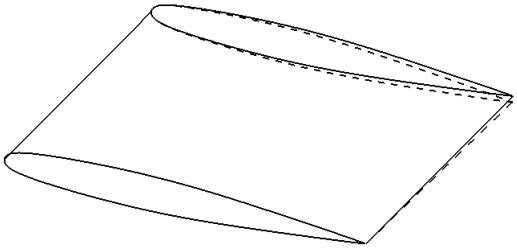


Figure 3.14 Comparison of CAD geometry before (solid line) and after twist (broken lines)

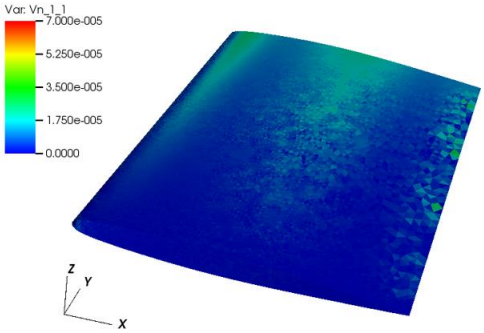


Figure 3.15 Comparison of difference between analytical and CAD based design velocity

3.3.3 Test Case 3

The next test case analyzed is the LS89 [88] axial high-pressure turbine nozzle guide vane designed at the Von Karman Institute for Fluid Dynamics (VKI). The geometry used in this work is described in [74] and is constructed by using Bézier curves to define the section side (SS) and pressure side (PS) as shown in Figure 3.16. The position of Bézier control points is defined relative to a camber line. The VKI in-house tool called "computer aided design and optimization tool for turbomachinery applications" (CADO) [89] is used as the CAD kernel for this application.

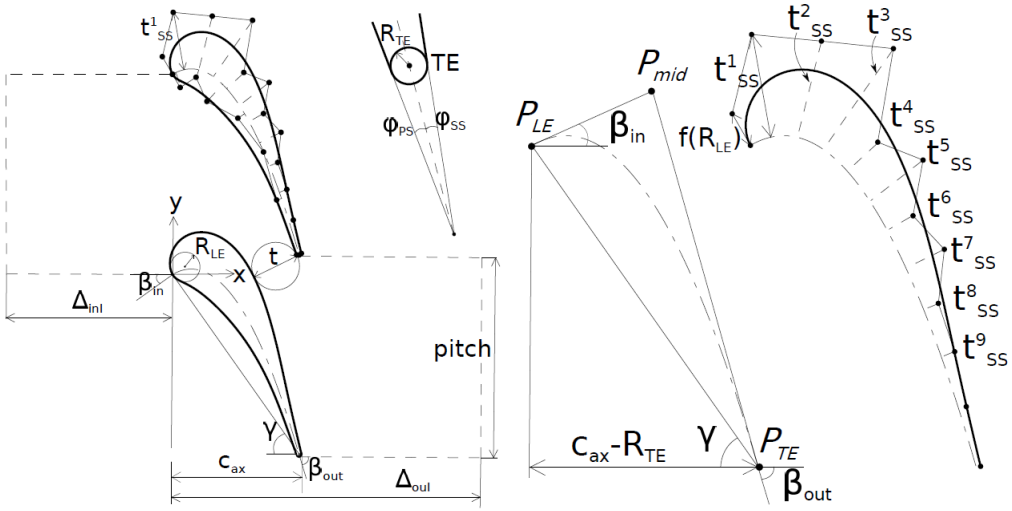


Figure 3.16 LS89 parameterization [74]

In this test case, design parameters are used which are relevant for the aerodynamic performance and are shown in Table 3.1.

Table 3.1 LS89 design parameters

parameters	symbol
axial chord length	C_{ax}
trailing edge thickness	D_{TE}
leading edge radius	R_{LE}
trailing edge radius	R_{TE}
PS thickness	$t_{PS}^1, t_{PS}^2, t_{PS}^3, t_{PS}^4$
SS thickness	$t_{SS}^1, t_{SS}^2, \ldots., t_{SS}^9$
inlet angle	β_{in}
outlet angle	β_{out}
PS wedge angle	φ_{PS}
SS wedge angle	φ_{SS}

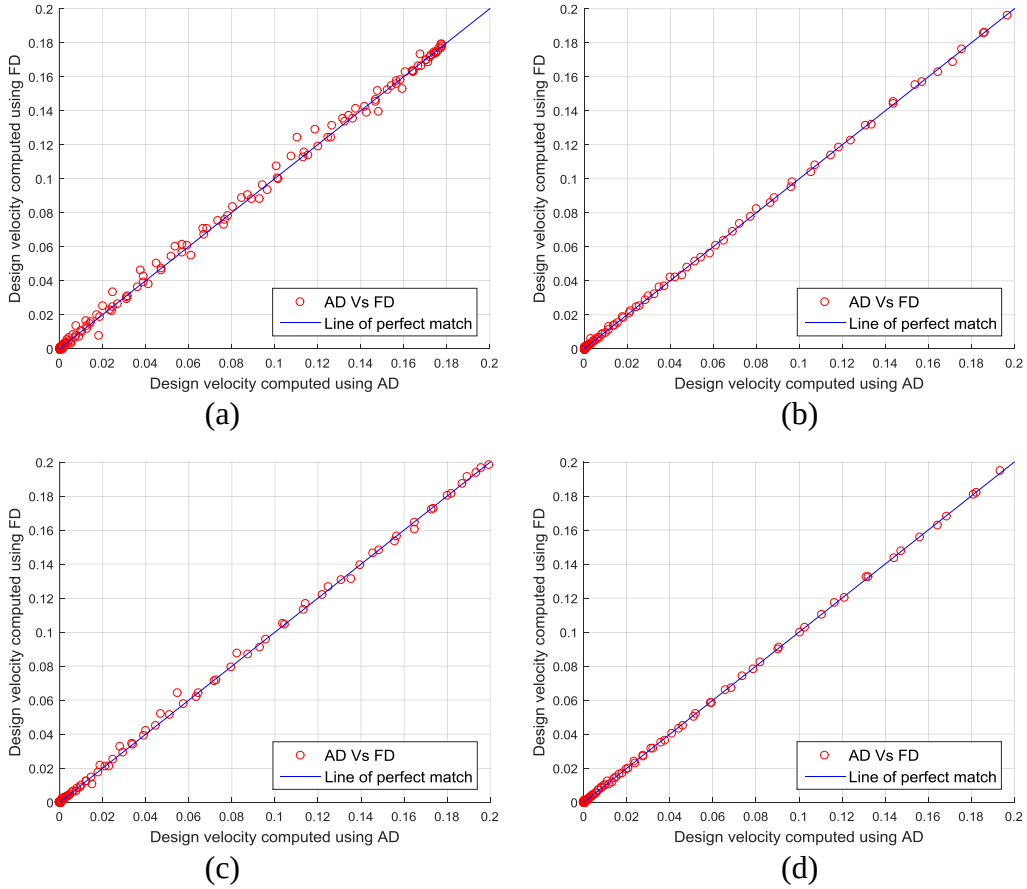


Figure 3.17 Design velocity comparison between the developed approach (FD) and AD for (a) LE Radius (R_{LE}), (b) SS thickness 1 (t_{SS}^1) , (c) SS thickness 4 (t_{SS}^4), (d) SS thickness 9 (t_{SS}^9)

The design velocities for the LS89 blade are computed using the approach presented in section 3.2 (referred as FD) and compared against the exact design velocities

computed by VKI by differentiating directly the CAD system using AD [74]. The results for four design parameters are shown in Figure 3.17, where a strong correlation between the two approaches is obtained (regression coefficient $R^2 > 0.995$), giving further confidence in the applicability of the developed approaches for the computation of design velocities.

3.4 Gradient computation

For a gradient based optimizer to establish a new search direction, it is necessary for the performance gradient (or sensitivity) to be evaluated with respect to each design variable as

$$\begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \cdots & \frac{\partial x_m}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial \theta_n} & \cdots & \frac{\partial x_m}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \frac{\partial J}{\partial x_2} \\ \vdots \\ \frac{\partial J}{\partial x_m} \end{bmatrix} \tag{3.11}$$

where n and m are the number of design variables and surface mesh points, respectively; J represents the current function of interest, be it the objective or constraint functions; θ represents the CAD feature parameter. The variables x_i represent the normal displacement of discrete points on the surface, the term $\partial x/\partial \theta$ represents design velocity and the third term $\partial J/\partial x$ represents the adjoint sensitivities. This provides flexibility for the use of different parameterization methods. For a CAD model, once the adjoint sensitivity (ϕ) and design velocity (V_n) are computed using approach outlined in this chapter, the matrix form of Eqn. 3.11 can be used to obtain the total change in objective function as the summation over the boundary as

$$\mathrm{dJ} = -\int_A \phi V_n dA\,. \tag{3.12}$$

where A represents the surface area of the boundary. Note that the minus sign in Eqn. 3.12 is due to the sign convention, where a positive adjoint sensitivity indicates that an outward movement of the model boundary would decrease the objective function. Knowing the change in objective function due to the parametric perturbation in question, the parametric sensitivity or gradient (S), can be calculated by normalizing this value with respect to the size of the parameter perturbation applied to the parameters as

$$S = \frac{dJ}{d\theta}. \tag{3.13}$$

The remaining sections of this chapter show the computation of gradients for two industrial CAD models created in a commercial CAD package SIEMENS NX, and their validation with respect to finite differences.

3.5 Validation of Gradients

In order to evaluate the developed approach for industrial geometry, gradients were calculated for a state-of-the-art nozzle guide vane of a high-pressure turbine [82, 90] and a rotor blade geometry [91] developed by Rolls-Royce Deutschland (RRD).

3.5.1 Nozzle Guide Vane (NGV)

A 3D CAD model of NGV geometry was built in Siemens NX. The NGV design defines the engine mass flow (and by association the turbine capacity, $\mathbb{Q}$), and is characterised by filleted ends with a cooling slot feature at the trailing edge. In this test case, the capacity is considered as the objective function, calculated as

$$\mathbb{Q} = \dot{m}\frac{\sqrt{T_t}}{p_t}, \tag{3.14}$$

where $\dot{m}$ denotes the inlet mass flow, T_t the total temperature and p_t the total pressure at the inlet using mass averaged values. This geometry is also investigated in [82, 90], where more details about the test case can be found.

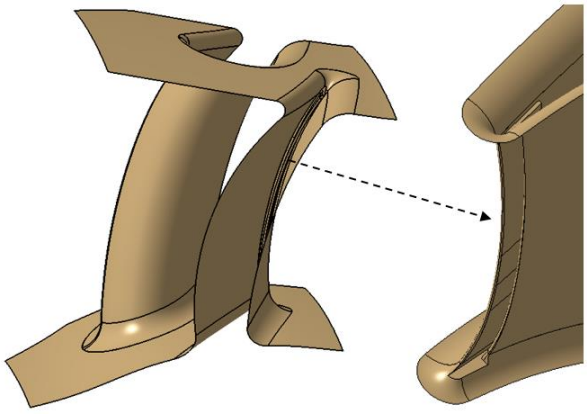


Figure 3.18 3D CAD model of NGV geometry in Siemens NX

Due to the symmetries in the model, the CFD simulations were conducted on one periodic section of the engine's annulus. Consequently, the sector domain shown in Figure 3.18 is used for both CFD and design velocity computation. The whole turbine model is constructed by parameterizing one blade, and subsequently using the total

number of blades to create a circular pattern of the blades to define the full turbine model. Thus, only altering the parameterization of one sectorial domain will automatically update the other blades in the full turbine model.

The CAD model was created as part of an automated in-house iSIGHT [92] workflow in RRD and twelve CAD parameters are analyzed. The parameters are perturbed using the same workflow. Figure 3.19 shows some of the CAD parameters considered in this test case, where *SS* represents the suction side and *PS* represents the pressure side of the blade profile.

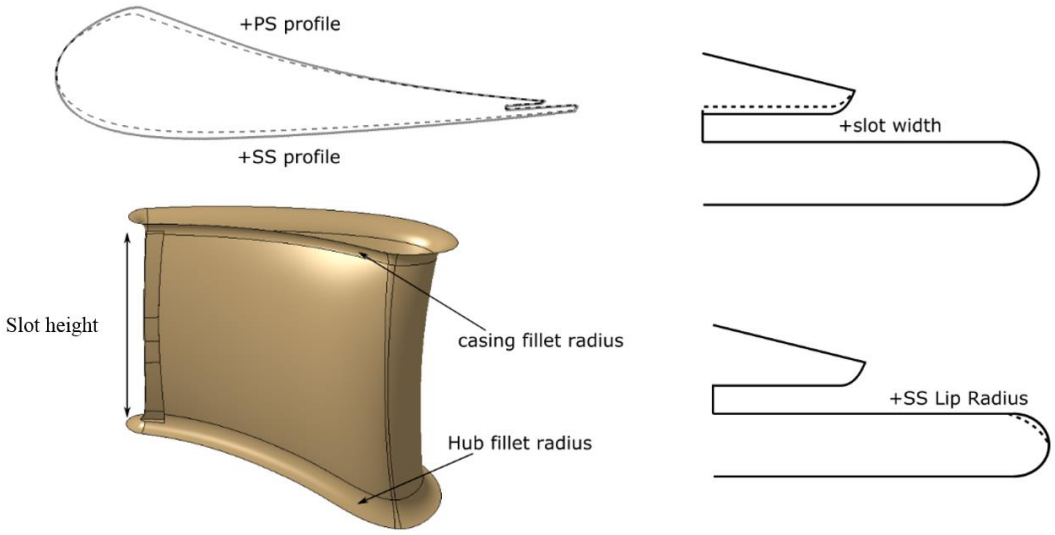
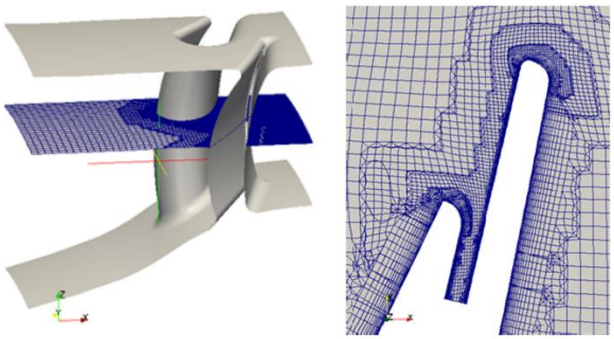


Figure 3.19 CAD feature parameters of NGV (not to scale)



(a) (b)

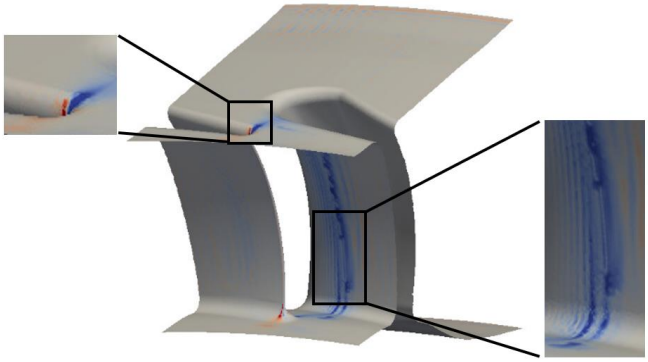
[1]Figure 3.20 (a) NGV CFD domain, and (b) mesh around trailing edge

For each design variable, a new geometry is created using the in-house tool with a perturbation step between 0.1% and 1% of the model size. The mesh for CFD analysis is then automatically created using BOXER meshing software [93] and contained

[1] Courtesy: Ilias Vasilopoulos, Rolls-Royce Deutschland, Germany

approximately 9 million nodes and 13 million cells. A typical example of the mesh is given in Figure 3.20, including a detailed view of the mesh around the trailing edge.

The primal and adjoint CFD results are obtained using the Rolls-Royce in-house CFD solver HYDRA. It is a nonlinear flow solver using a node-based finite-volume discretisation method and a pseudo-time-marching scheme to reach steady state, accelerated by a block-Jacobi preconditioner and a geometric multigrid technique. HYDRA has been successfully applied on industrial test cases [81, 82], and more details on the underlying theory and implementations can be found in [9, 83].



[2]Figure 3.21 NGV Adjoint sensitivity map

Herein, the steady state RANS equations are solved with the Spalart-Allmaras turbulence model and wall functions and its corresponding discrete adjoint solver. The convergence criteria used here required the residual to reduce by nine and five orders of magnitude for the primal and adjoint solutions, respectively. The adjoint sensitivity map is illustrated in Figure 3.21 in which areas of extreme sensitivity are shown in red and blue, representing areas where the boundary of the model should be displaced outwards or inwards respectively, to achieve an increase in objective function. Areas of low sensitivity (shaded in grey) show that the objective function is not sensitive to the movement of the boundary in those areas.

For each design parameter, a design velocity field was calculated using the approach presented in this chapter and linked with the adjoint sensitivity maps. The required surface facets are created in GMSH using Delaunay triangulation algorithm with the target element size to be 0.5% of the global model size. The design velocity contours for the casing fillet, *SS* profile and hub fillet are shown in Figure 3.22. Finally, the

[2] Courtesy: Ilias Vasilopoulos, Rolls-Royce Deutschland, Germany

change in performance caused by each parametric perturbation is predicted by taking the inner product of the sensitivity map with the corresponding design velocity field, using Eqn. 3.12. Figure 3.23 compares the gradients obtained using the adjoint approach and those calculated using finite differences.

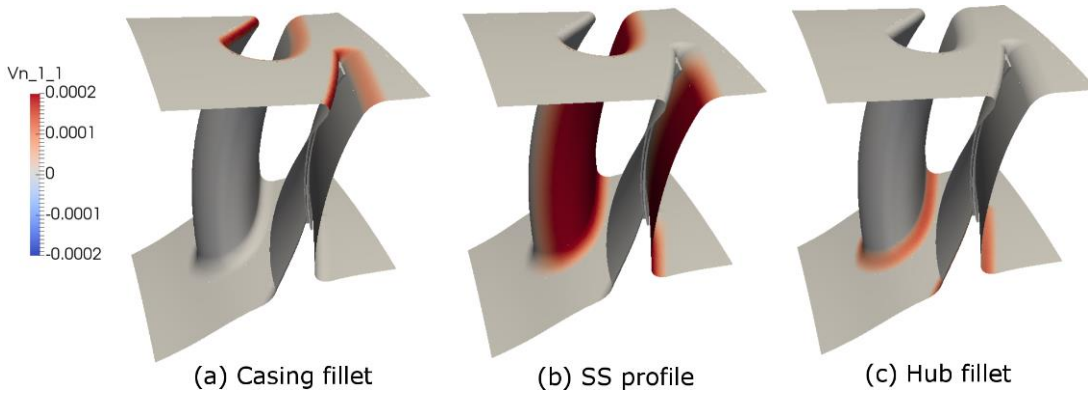


Figure 3.22 Design velocity contours for NGV

It should be noted that the parameters considered here are controlling the shape of the trailing edge of the NGV, which is characterized by highly non-linear flow. So, the difference in the magnitudes of gradients ($\sim 50-60\%$.) may be related to the step size used in the finite-differences, or inaccuracies in the adjoint solution near the trailing edge discontinuity. As the CAD model geometries were created as part of an automated industrial workflow it was not possible to perform finite difference analyses for different step sizes.

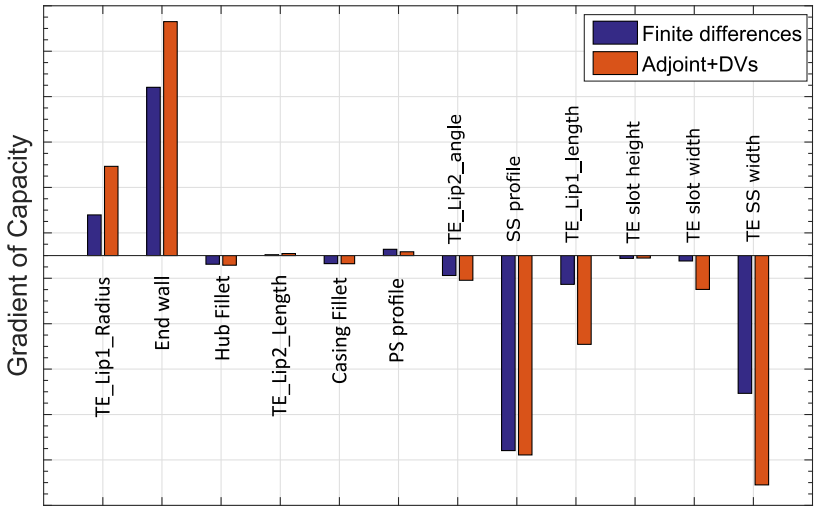


Figure 3.23 Validation of gradient of capacity predicted by adjoint results for NGV

The important point to be noted is that all gradients have the same direction and is reliable enough to drive a gradient based optimization method. Also, Rolls-Royce were unable to achieve results as accurate as the ones presented for this test case, using any

other approach. Note the values of gradients are omitted due to confidentiality restrictions by RRD.

3.5.2 Turbine Rotor Blade model

The rotor blade geometry is shown in Figure 3.24, where a 3D parametric CAD model is created in Siemens NX. The key feature of this blade is its winglet, a detailed view of which is included in Figure 3.24. This feature is used to reduce the leakage mass flow, which leads to an increased rotor efficiency by decreasing the driving pressure difference across the rotor's tip. A detailed study on winglets and squealers has been conducted in [91]. In this work, the study is focused on the winglet geometry and total pressure loss between inlet and outlet is considered as the objective function. A total of seven CAD parameters controlling the winglet geometry are considered as design variables.

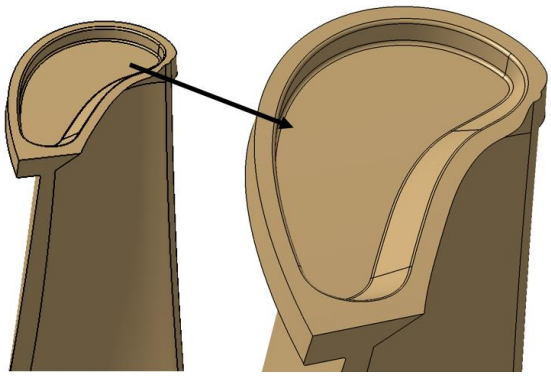
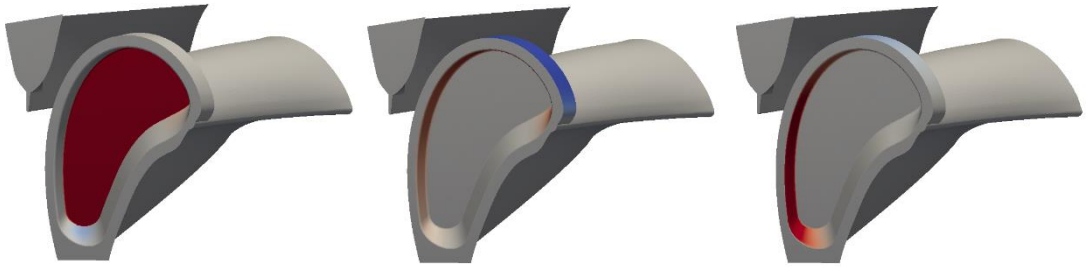


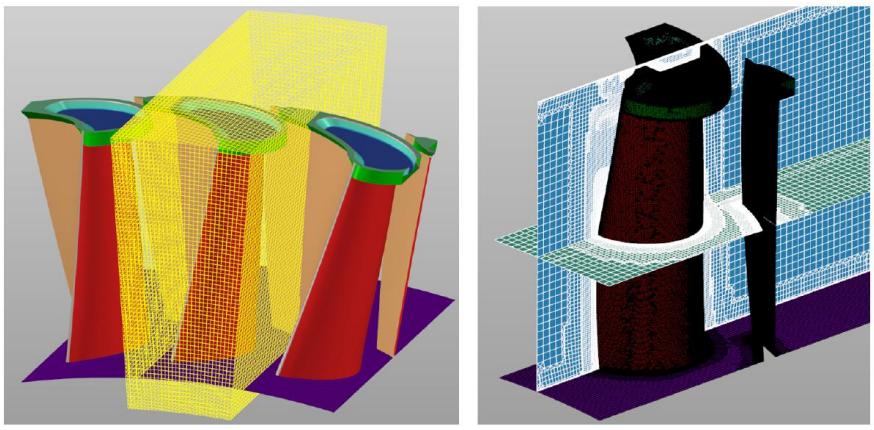
Figure 3.24: Rotor Blade CAD model in Siemens NX

The design velocity contours for three of the parameters are shown in Figure 3.25. The CFD mesh is generated in BOXER where a periodic sector is created from the initial geometry, as illustrated in Figure 3.26(a). The produced CFD domain of the baseline geometry is shown in Figure 3.26(b) and has around 9 million nodes.



(a) Cavity height (b) SS front offset (c) SS rear angle primary

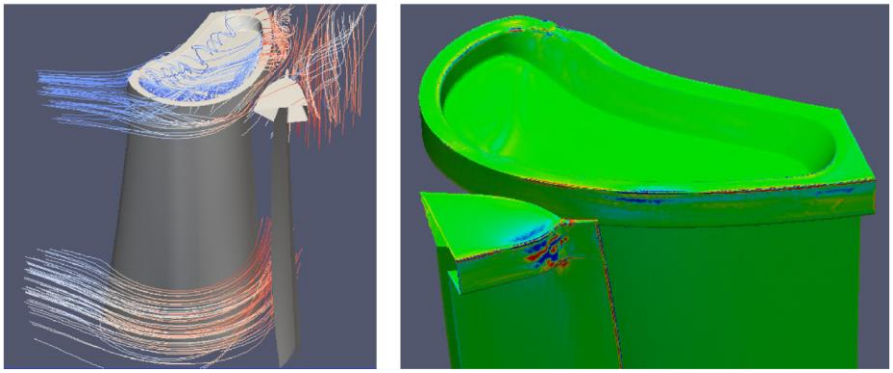
Figure 3.25: Design velocity contours for turbine rotor blade



(a) (b)

[3]Figure 3.26: BOXER mesh generation: a) periodic section of initial geometry and b) rotor CFD domain

The primal and adjoint solutions were obtained from the HYDRA solvers, by using the steady RANS solver with a one equation Spalart-Allmaras turbulence model and wall functions. The converged flow for the baseline geometry is shown in 3Figure 3.27(a), where it can be seen that the fluid stays inside the winglet and re-circulations occur which may cause an increase in total pressure losses within the tip gap. The corresponding adjoint sensitivity map is illustrated in 3Figure 3.27(b).



(a) (b)

[3]Figure 3.27 Flow and adjoint solution: a) Streamlines coloured by relative velocity magnitude and b) Sensitivity map focused on winglet.

The gradient of the objective function (pressure loss) with respect to design parameters obtained using the approach presented in this chapter, is compared with the corresponding finite difference value in Figure 3.28. The adjoint method combined with design velocities underpredict the gradient values by approximately 60% for the

[3] Courtesy: Ilias Vasilopoulos, Rolls-Royce Deutschland, Germany

"SS Rear offset" parameters, while for all other parameters the difference is between $5-10\%$. It is to be noted that the values of gradients are omitted due to confidentiality restrictions from RRD.

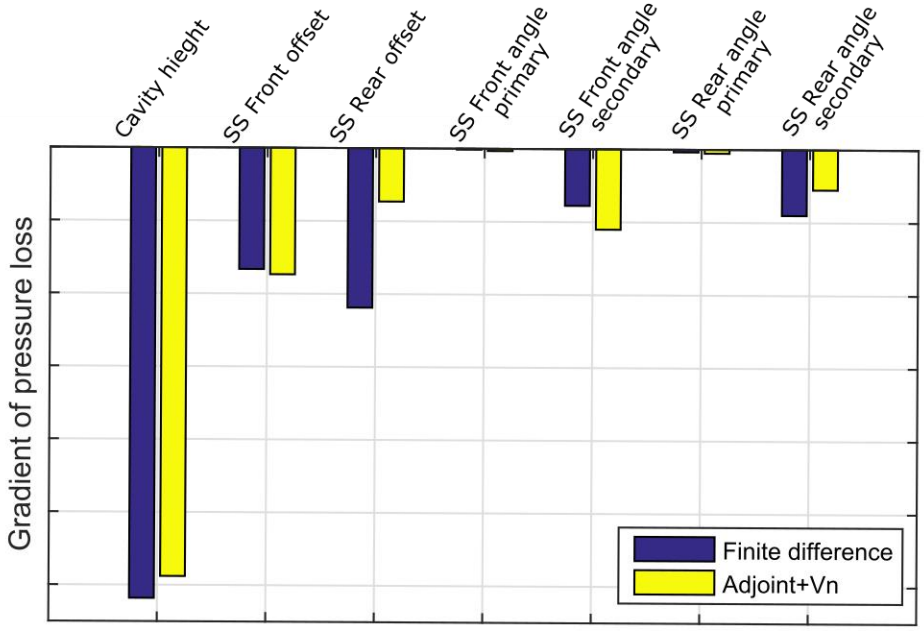


Figure 3.28: Validation of gradient predicted by adjoint results for rotor blade

3.6 Summary

The main objective of this chapter was to develop an automated workflow (see Figure 3.6) to calculate the design velocity with respect to the parameters which define the shape of a feature-based CAD model. In this regard, a robust approach was presented which uses the discrete representation (facets) for the CAD model boundary. The computed design velocities were validated against analytical calculations.

The design velocities were linked with adjoint sensitivities to facilitate the computation of performance gradients and was applied on industrial size test cases. The approach was demonstrated on three test cases, where the computed design velocities were compared against the analytical formulations of boundary movements. Later, the design velocities were linked with adjoint surface sensitivities to output the gradients of performance with respect to CAD parameters. The accuracy of the computed gradients was compared against the finite difference results for two turbomachinery components built in the SIEMENS NX CAD system and two aerodynamic surfaces built in CATIA V5. The flow solution for the test cases was obtained using the CFD solvers HYDRA and SU2.

In terms of computational efficiency, it was shown that calculating design velocities was computationally inexpensive compared to CFD analyses and enhances the ability of adjoint methods to reduce the optimization time for industrial test cases with large number of design parameters. In terms of computational requirement, for the industrial models one CFD analysis took approximately 24 hours using 20 cores, consequently the finite difference approach using the twelve design variables requires approximately two weeks of computational effort. The adjoint approach took two days, solving one primal and one adjoint solution. The computation of design velocities is carried out in parallel with the flow analysis and required approximately 30 minutes for all parameters on a 3.60GHz workstation with 16GB RAM. The overall reduction in computational time is therefore 12 days for all 12 parameters. This saving would further increase with the number of parameters. In the next Chapter, the methodologies developed here will be used to present an automated workflow for optimization of parametric CAD models created in the commercial CAD system CATIA V5.

Chapter 4 CAD-based adjoint optimization

4.1 Sequential least square programming (SLSQP)

The SLSQP method belongs to the class of Sequential Quadratic Programming (SQP) method used for solving nonlinearly constrained optimization problems. Consider an optimization problem defined as

$$\begin{aligned} Minimize: &\quad J(\boldsymbol{\theta}), \\ Subject\ to: &\quad h(\boldsymbol{\theta}) = 0, \\ &\quad g(\boldsymbol{\theta}) \leq 0, \end{aligned} \tag{4.1}$$

where $J(\boldsymbol{\theta}), h(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta})$ are the function to be minimized, equality and in-equality constraint respectively. Now, using the standard Lagrangian formulation, a new function $L(\boldsymbol{\theta})$ can be formulated as

$$L(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda_e h(\boldsymbol{\theta}) + \lambda_i g(\boldsymbol{\theta}), \tag{4.2}$$

where λ_e and λ_i are the Lagrangian multipliers for equality and inequality constraint respectively. For the active inequality constraint, the constraints satisfy $g(\theta) = 0$. A basic SQP algorithm can be used to obtain a search direction to minimize the objective function in presence of constraints.

In SLSQP, the Lagrangian multiplier are calculated through least-square method. Assume a single constraint problem with only equality constraint, the Lagrangian function can be written as

$$L(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda_e\, h(\boldsymbol{\theta}). \tag{4.3}$$

Now, as the design reaches the optimum the derivative of Lagrangian function should be close to 0 as

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \frac{\partial J}{\partial \boldsymbol{\theta}} + \lambda_e \frac{\partial h}{\partial \boldsymbol{\theta}} \approx 0. \tag{4.4}$$

With the gradient of objective function ($\partial J/\partial \boldsymbol{\theta}$) and constraint ($\partial h/\partial \boldsymbol{\theta}$) computed, λ_e may not exist such that Eqn. 4.4 is close to 0 for all the design parameters ($\boldsymbol{\theta}$). So, the multiplier λ_e can be calculated using least-square method as:

$$\lambda_e = -\left[\left(\frac{\partial h}{\partial \theta}\right)^T \left(\frac{\partial h}{\partial \theta}\right)\right]^{-1} \left(\frac{\partial h}{\partial \theta}\right)^T \left(\frac{\partial J}{\partial \theta}\right) \tag{4.5}$$

The advantage of using SLSQP algorithm is that it avoids the need for determining the weighting factors for including constraints directly into the optimization problem and are approximated by using least-square formulation.

4.2 Automated optimization framework

In this chapter, the methodologies developed in the preceding chapter for computing performance gradients with respect to CAD parameters, are used for the optimization of flow problems. The CAD models are created in CATIA V5 and optimized using the SLSQP algorithm implemented in Scipy [94]. Here, a gradient based optimization methodology is followed to minimize a function with any combination of bounds, equality and inequality constraints. The key inputs to the Scipy SLSQP optimizer are

- *func:* function to evaluate the value of objective function.
- *x0:* set of design variables.
- *bounds:* lower and upper bound for parameters. If the input to the bounds is *None*, then the optimizer considers the parameters to be bounded as $(-\inf, \inf)$.
- *fprime:* function to evaluate the gradient of objective function.
- *f_eqcons:* function to evaluate the equality constraint.
- *fprime_eqcons:* function to evaluate the gradient of equality constraint.
- *Ieqcons:* function to evaluate the in-equality constraint.
- *fprime_ieqcons:* function to evaluate the gradient of in-equality constraint.
- *iter:* maximum number of optimization runs.

An automated optimization framework (see Figure 4.1) is developed to link different modules created for CAD parameter perturbation, mesh generation, design velocity calculation, CFD flow analysis etc. The interface between the CAD modelling system CATIA V5 in Windows 10 and the CFD flow solvers in Linux was established through the Dropbox Cloud network [95].

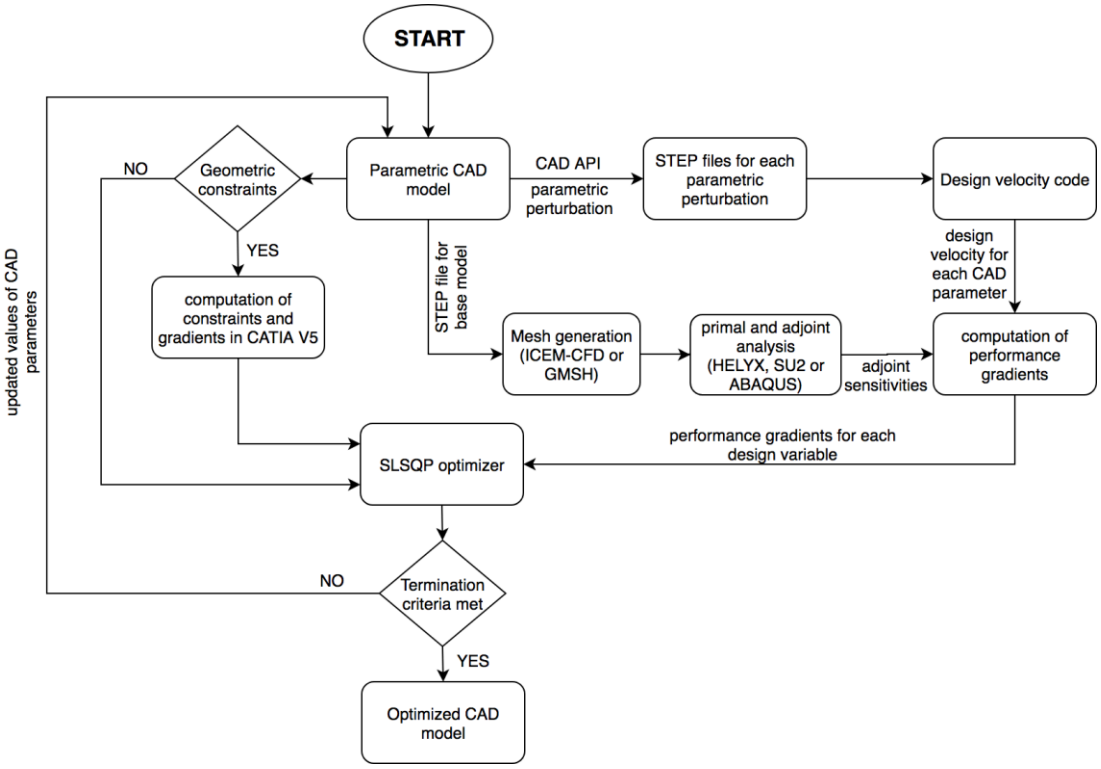


Figure 4.1 Flowchart depicting the automated optimization process

4.3 NACA0012 aerofoil

In order to demonstrate the developed approaches, the first test case considered is the benchmark two-dimensional NACA0012 aerofoil [96, 97]. NACA0012 represents a symmetrical aerofoil with zero chamber and 12% thickness to chord ratio.

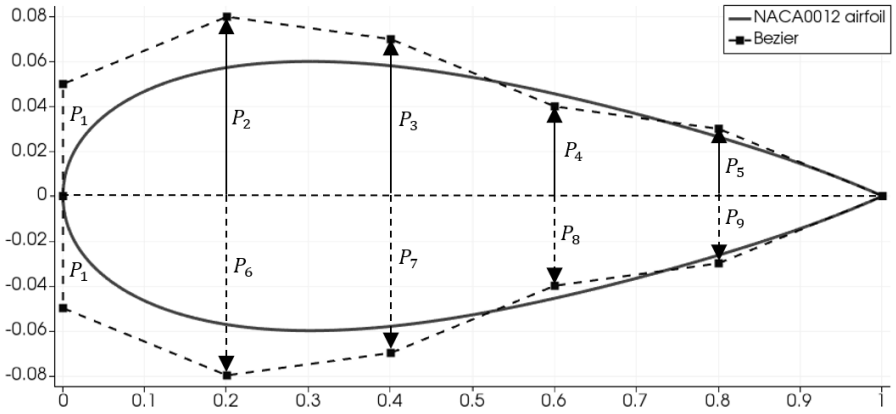


Figure 4.2: NACA0012 aerofoil with Bezier control points

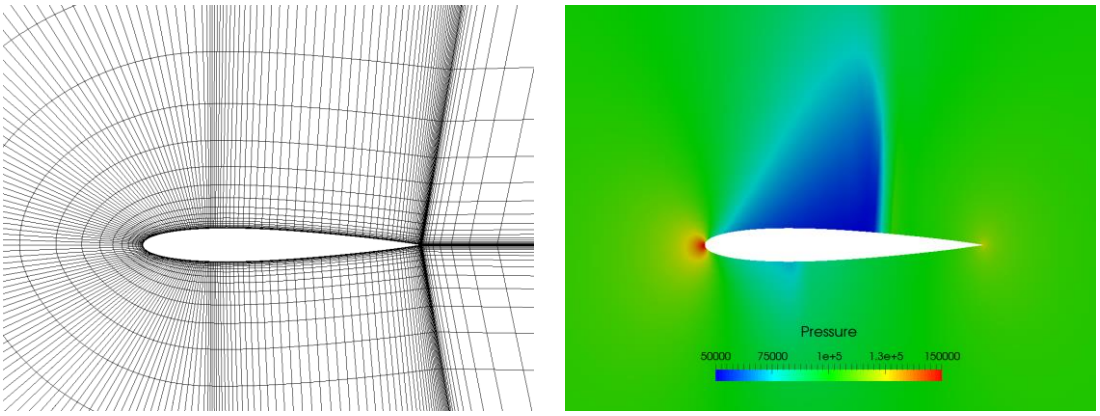
Here, the aerofoil is constructed using two Bézier curves (one top and one bottom), each with seven control points adhering with the following constraints: the leading edge and trailing edge points are fixed, and the first control point on each surface after the leading edge is constrained to move by equal amounts in opposite vertical offsets

from the leading-edge point. This is to preserve C_2 continuity at the leading edge. All the control points only vary in the vertical direction. The initial geometry with Bezier control points is shown in Figure 4.2.

The flow conditions are defined as:

- Freestream Temperature = 273.15 K
- Freestream Mach number = 0.80
- Angle of attack (α) = 1.25^o
- Objective Function = $min(C_D)$
- No. of design variables = 9 (shown in Figure 4.2)

The CFD mesh is created in ICEM-CFD [98] using a multi-block strategy with 200 points defining the aerofoil and 31 points in the direction normal to the aerofoil as shown in Figure 4.3(a). The mesh is exported in CFD General Notation System (CGNS) format to be used within SU2. A strong shock is formed over the aerofoil as shown in Figure 4.3(b) leading to an increased drag.



(a) (b)

Figure 4.3 (a) Mesh around the NACA0012. (b) Pressure flow field

The adjoint sensitivities obtained for reducing drag are shown in Figure 4.4. For two-dimensional (2D) test cases, a 3D CAD model is constructed by extruding the 2D profiles by $\pm 1mm$ in the direction normal to the plane as shown in Figure 4.5. Now the design velocities are computed using the process outlined in Chapter 3 and linked with adjoint sensitivities to compute the required gradients.

To validate the gradient values produced using the developed framework, the adjoint sensitivities and CAD design velocities were coupled and compared against the gradients calculated by finite differences (FD) using the analysis for geometries with parametric perturbations of 0.01c (where c defines the chord length). The results are

shown in Figure 4.6, a strong correlation between the two sets is clear, giving confidence in the implemented process.

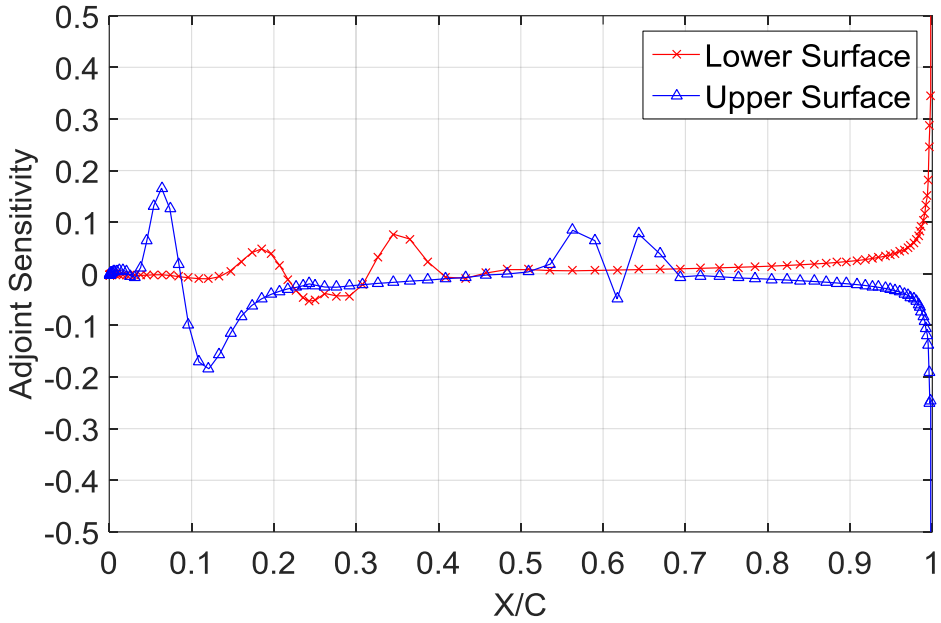
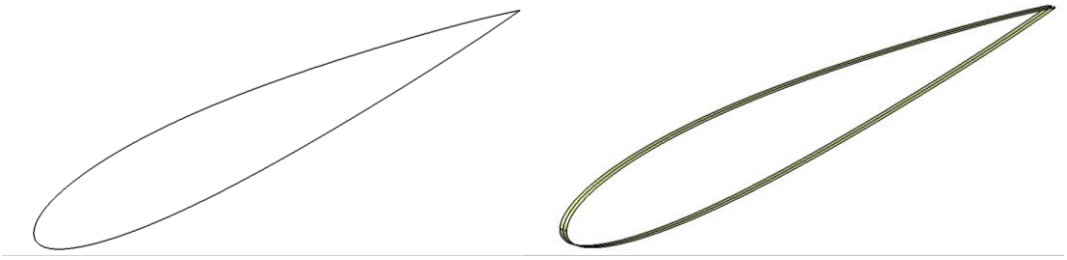


Figure 4.4 Adjoint surface sensitivity on NACA0012 with drag as objective function



(a) (b)

Figure 4.5 (a) 2D aerofoil geometry, (b) 3D CAD model constructed by extruding the aerofoil profile

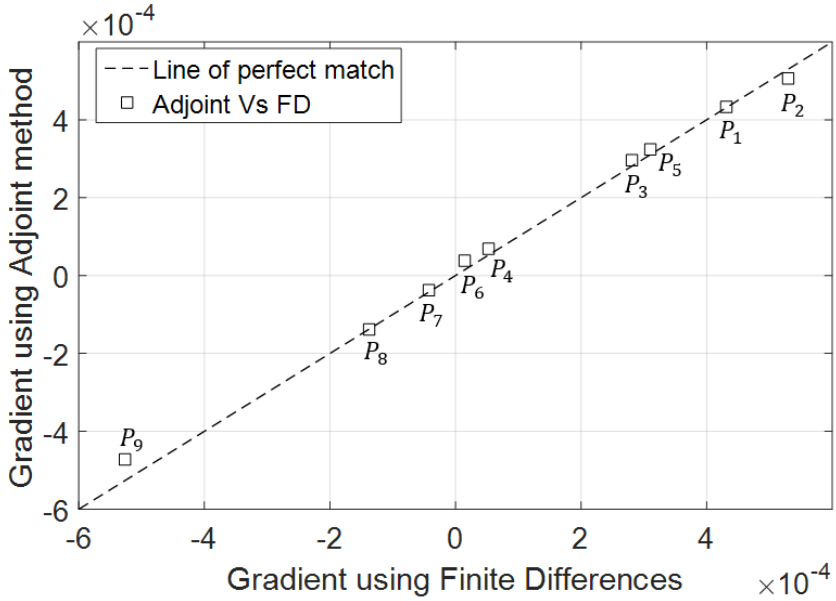


Figure 4.6 Gradient Validation for NACA0012 with drag as objective function

Un-constrained optimization

An un-constrained shape optimization problem is set-up using CAD parameters with parametric bounds as design variables. For each optimization iteration, a new multi-block CFD mesh is created in ICEM CFD using an automated blocking script. The optimization process led to the reduction in drag coefficient (C_d) from 0.0215053 to 0.00065. A comparison between the original and optimized aerofoil geometries along with the corresponding pressure coefficient (C_p) values is shown in Figure 4.7 and the optimization history in Figure 4.8. It should be noted that the ideal solution for this problem is an airfoil which looks like a "flat plate" but using the parameter bounds restricts the optimizer to result in such geometry.

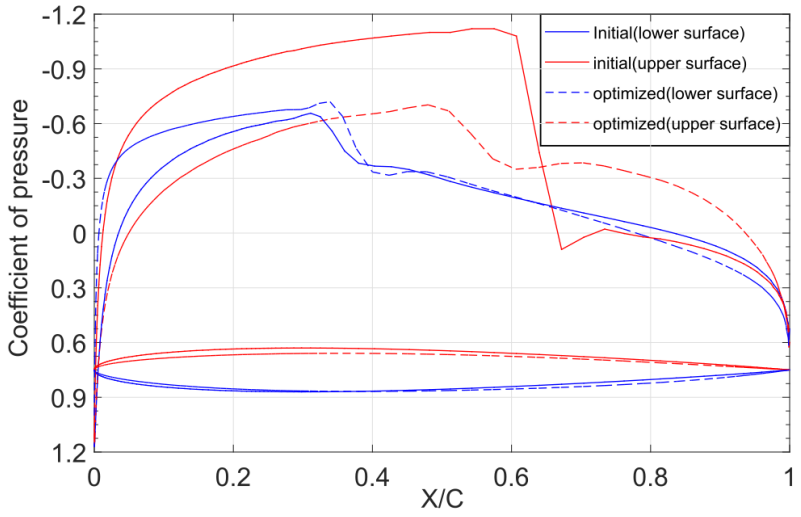


Figure 4.7 C_p and shape comparison of initial (solid lines) and optimized (dashed lines) NACA0012 aerofoil

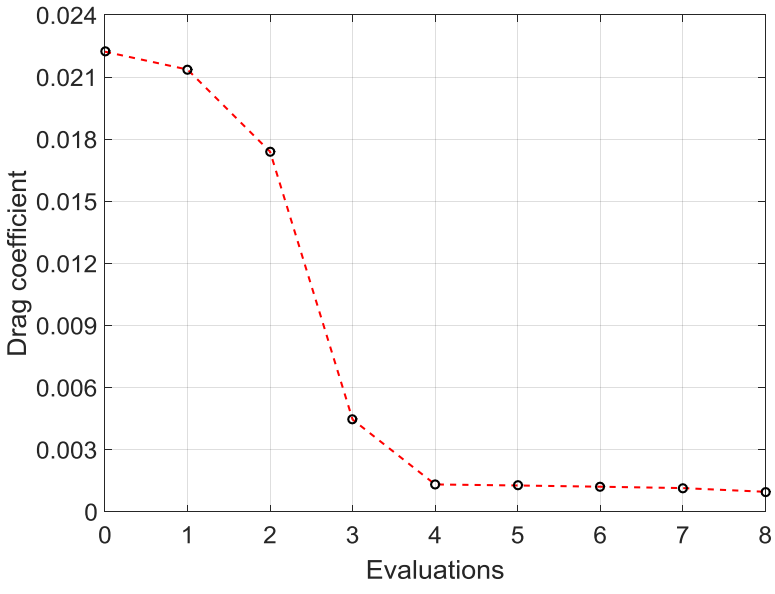


Figure 4.8 Optimization history for NACA0012 (un-constrained)

Constrained optimization

Following the benchmark problem of NACA0012 with constraint definition [97], constraints are imposed on the aerofoil such that the thickness at any section of the aerofoil should not be less than that at the start of optimization. This is achieved within the CAD system, by inserting 18 cross-section planes between the leading and trailing edge of the aerofoil and computing the vertical clearance between the upper and lower aerofoil curves at these locations as shown in Figure 4.9. Now, the optimization of the aerofoil is performed in transonic flow regime ($M = 0.80$), with zero angle of attack ($\alpha = 0^o$) resulting in a symmetrical flow around the aerofoil as shown in Figure 4.10(a).

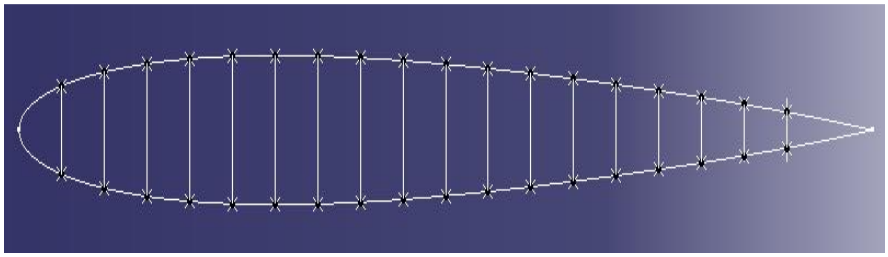
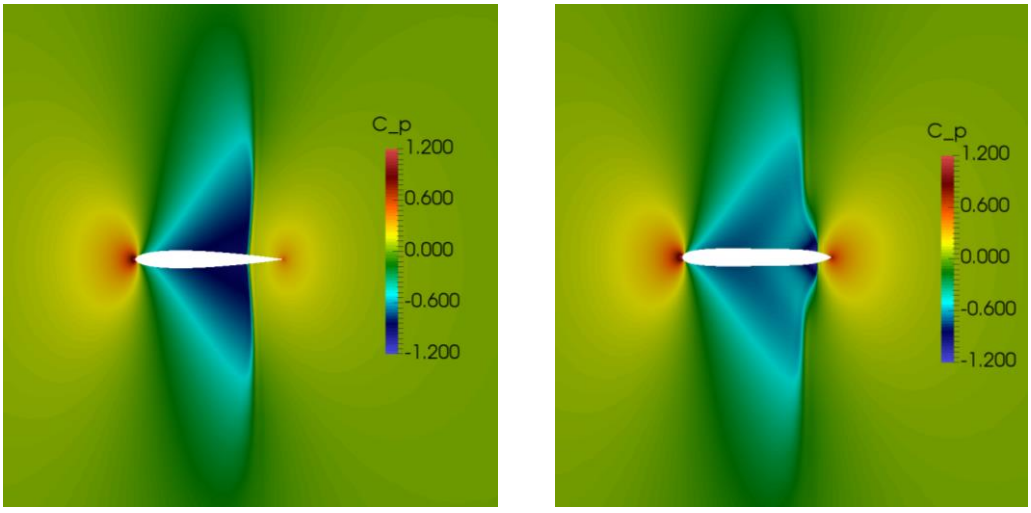


Figure 4.9 NACA0012 CAD model with thickness computation

The aerofoil thickness calculated at different sections is used as the input value of the constraint supplied to the optimizer. This involved formulating a function that computes the difference between the length of these 18 lines with the original length and used as an input to the function *Ieqcons.* The gradient of these constraints is then calculated in a finite difference manner by perturbing each parameter by a small amount and using it as an input to the function *fprime_ineqcons.*



(a) (b)

Figure 4.10 pressure contours on NACA0012 (a) initial, (b) optimized with thickness constraint

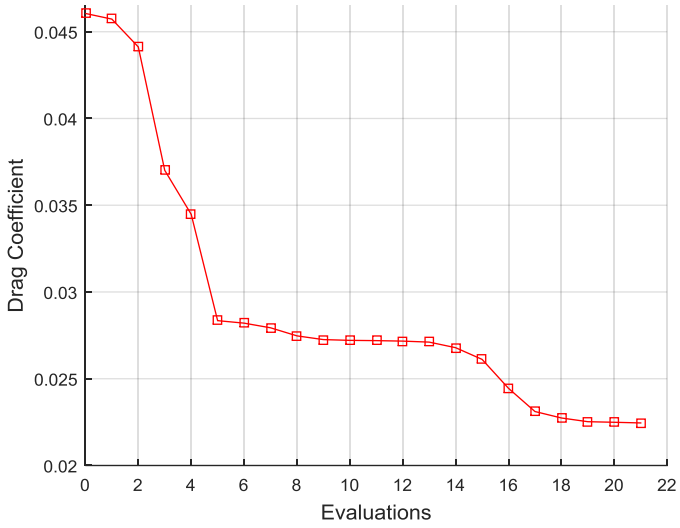


Figure 4.11 Optimization history for NACA0012 (thickness constraint)

The results of the optimization process showed a reduction in the value of C_d from 0.046054 to 0.02245 in 21 optimization steps as shown in Figure 4.11. The pressure distribution on the optimized aerofoil is shown in Figure 4.10(b) where the shock is seen to be weakened resulting in a reduced drag. The comparison between the initial and optimized geometry is shown in Figure 4.12. Please note that similar results were presented for the optimization of NACA0012 in [96].

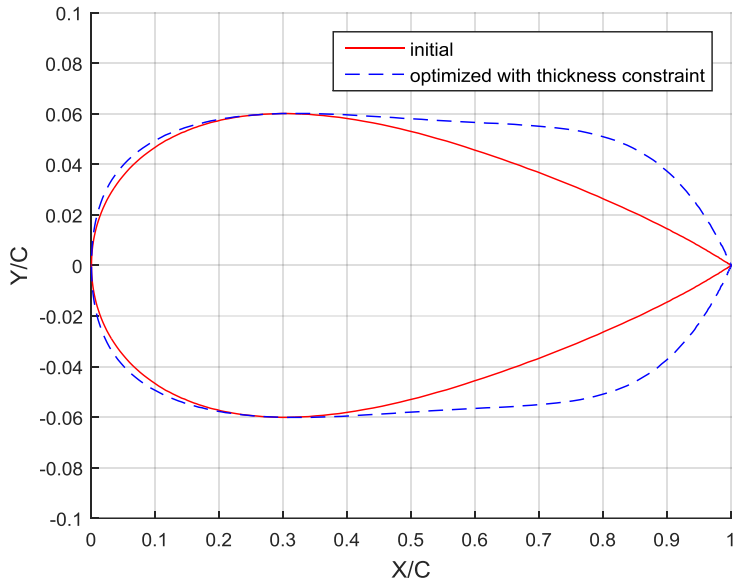


Figure 4.12 NACA0012 aerofoil optimized with thickness constraint

4.4 ONERA M6 Wing

The next optimization test case considered is the standard ONERA M6 wing model [99], which is a swept, semi-span wing with no twist. A parametric CAD model for the wing was constructed, using four different cross-sections along the wing span.

Each cross-section is defined using the same parameterization strategy as was used for NACA0012. In order to keep the shape of the wing tip fixed, only the first three sections were parameterized resulting in a total of 27 CAD parameters. The wing is then constructed by sweeping a surface through these section curves as shown in Figure 4.13.

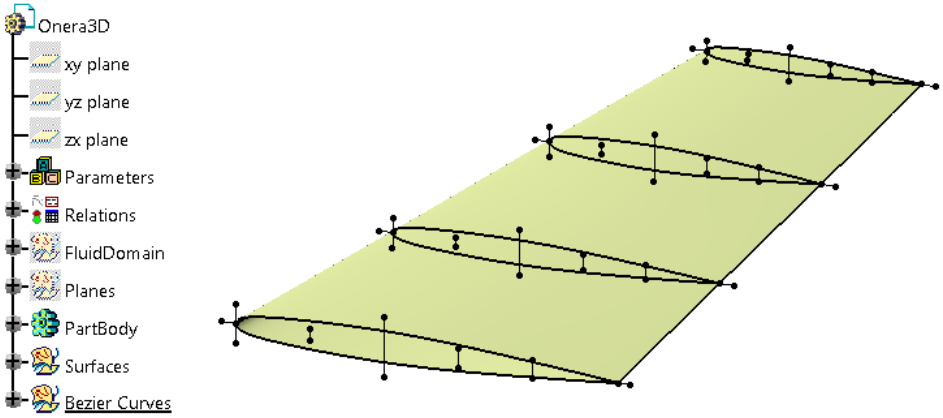


Figure 4.13 ONERA M6 CAD model showing Bézier control points for section profiles

The following flow conditions are defined for CFD analysis:

- Freestream Temperature = 288.15 K
- Freestream Mach number = 0.8395
- Angle of attack (α) = 3.06°
- Objective Function = $min(C_D)$
- No. of design variables = 27

An unstructured mesh was created in GMSH with 154,617 nodes and 707,115 tetrahedral elements, the respective surface mesh is shown in Figure 4.14 and used for both primal and adjoint analysis. The mesh density near the leading and trailing edge of the wing are controlled by implementing a background mesh field with refinement boxes. The pressure flow field in Figure 4.15 shows the formation of shock on the upper surface of the wing, which is similar to results available from the literature [36, 100, 101]. The convergence of the residuals of the density and correspondent adjoint variable equations for the initial wing is shown in Figure 4.16.

The design velocity is calculated using the work described in Chapter 3. All parameters were perturbed by 1% of the wing chord length, and the surface facets for computation of design velocities are created following the same strategy used for creating the CFD mesh. The generation of 28 STEP files took 21s, while the creation of surface facets

in GMSH took 1140s. The design velocity computation was completed in 200s, leading to the total process time of 1361s. The creation of facets was later parallelized (4 cores) to reduce the total process time to 468s.

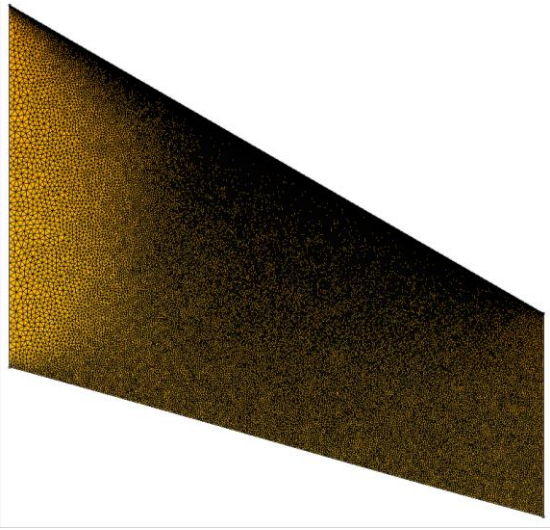


Figure 4.14 ONERA M6 CFD mesh

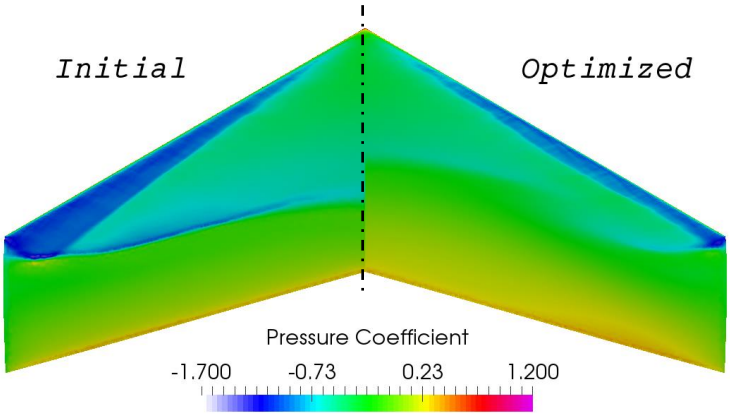


Figure 4.15 Pressure contours for initial and optimized ONERA M6

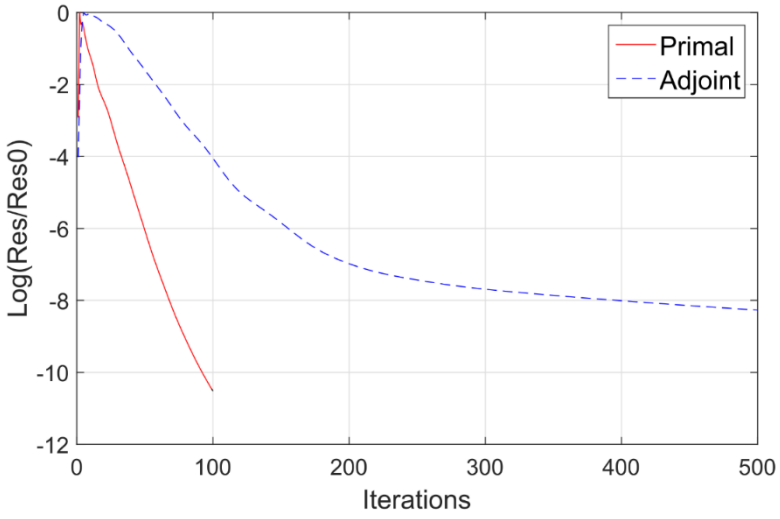


Figure 4.16 Residual convergence for initial ONERA M6

The contours for six of the parameters controlling the upper surface of the wing is shown in Figure 4.17. The gradient of performance (i.e. drag) with respect to CAD parameters is compared to those calculated using finite differences and is shown in Figure 4.18.

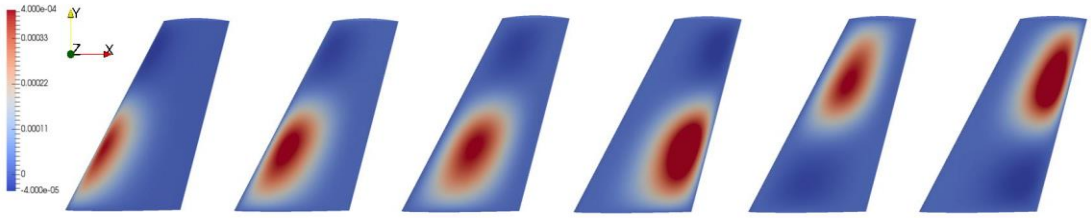


Figure 4.17 Design Velocity contours for ONERA M6

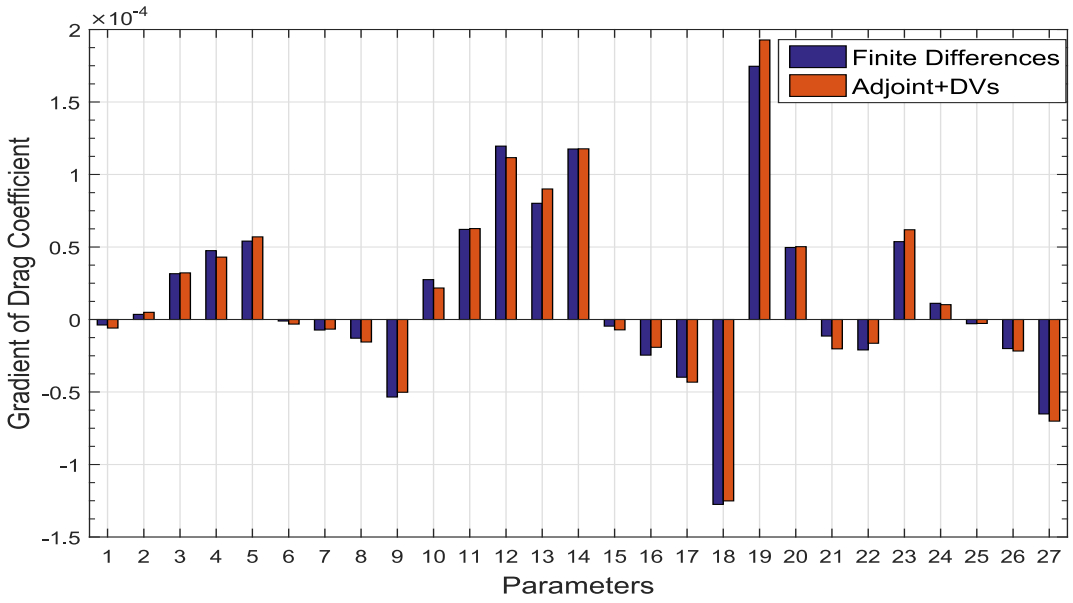


Figure 4.18 Validation of gradient of drag to CAD parameters predicted by adjoint results for ONERA M6 wing

For each parameter perturbation, a new mesh is generated with sufficient and similar density in an attempt to maintain the grid independence of the results, hence limiting the distortion of the finite difference calculations. The maximum differences between the values of gradients computed using the approach presented in Chapter 3 and those obtained using finite difference is less than 10% for most of the design parameters.

Un-constrained optimization

The performance gradients calculated for each CAD parameter are used for an un-constrained optimization, leading to a reduction in C_d from 0.012135 to 0.00303 in 12 iterations as illustrated in Figure 4.19. A comparison of the pressure coefficient

between the initial and optimized geometry at two different cross-sections is shown in Figure 4.20.

During the optimization process, a reduction of thickness at the leading edge is observed and the point of maximum camber moves slightly aft, resulting in a weakened system of shocks or the elimination of the rear shock, as observed at 60% span. Note as well that the lift is significantly reduced by increasing the aft camber. Both reductions in shock strength and lift contribute to limit the total drag produced.

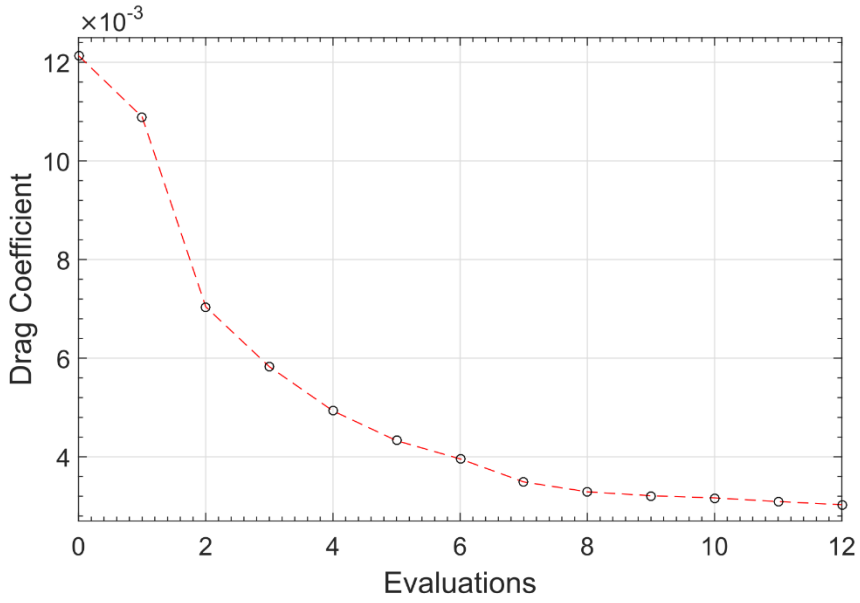
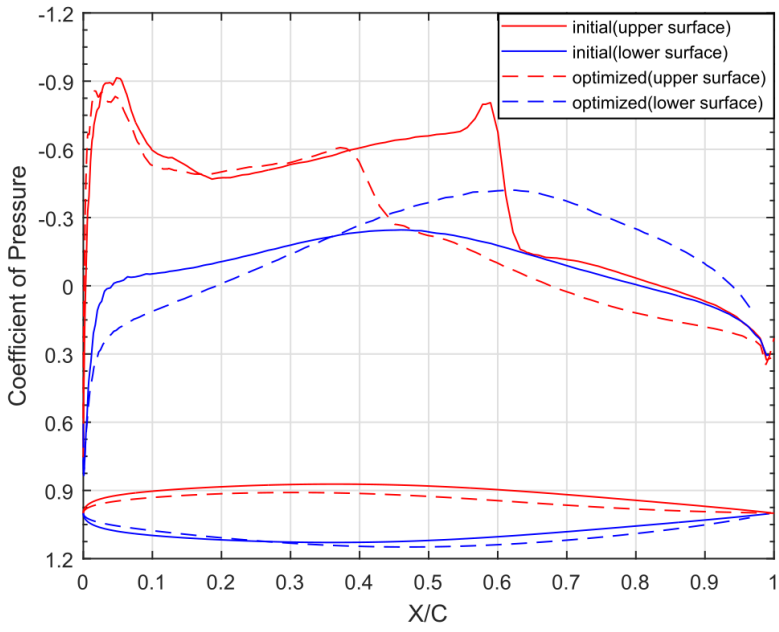
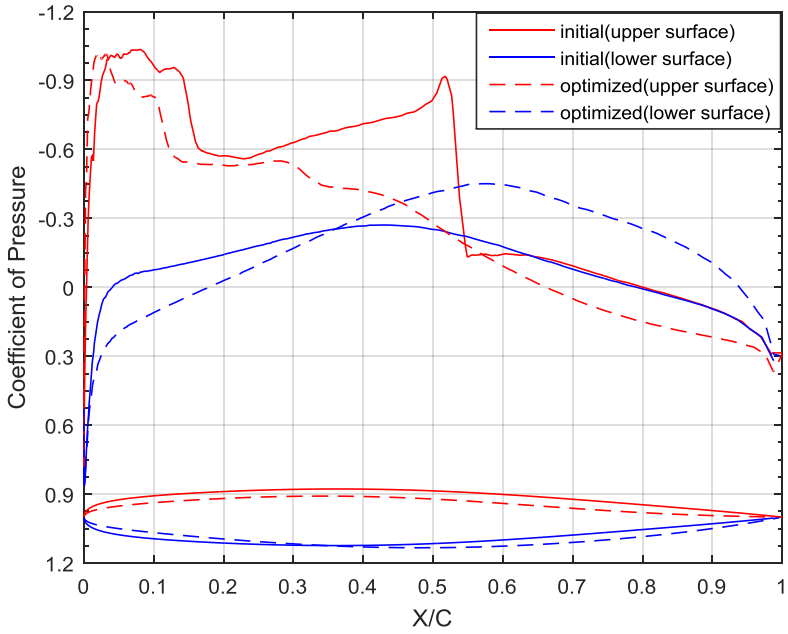


Figure 4.19 Un-constrained optimization history for minimizing drag on ONERA M6



(a)



(b)

Figure 4.20 C_p distribution along the wing (a) 30% span, (b) 60% span

Constrained optimization

The constrained optimization of the ONERA M6 wing is performed with the objective to minimize the drag, while constraining the lift to be above certain value (i.e. $C_l = 0.2864$). The gradient of the lift constraint with respect to CAD parameters was calculated by performing one additional adjoint analysis in SU2, with lift as the objective function. The optimization results can be visualized in Figure 4.21, where pressure contours on the initial and optimized wing are shown.

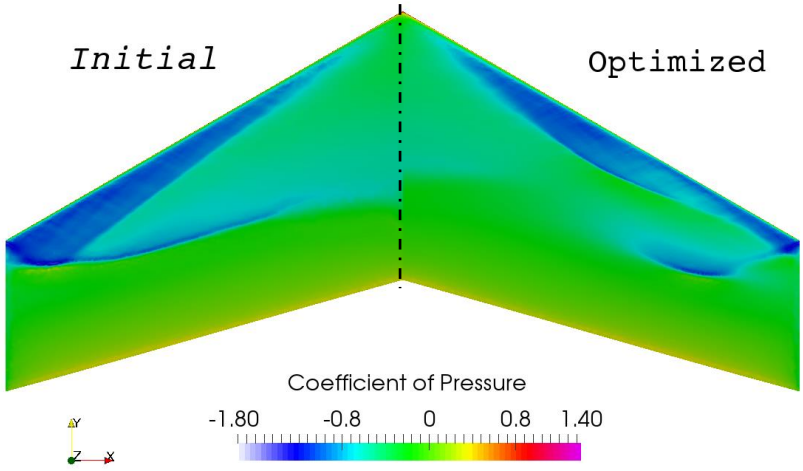
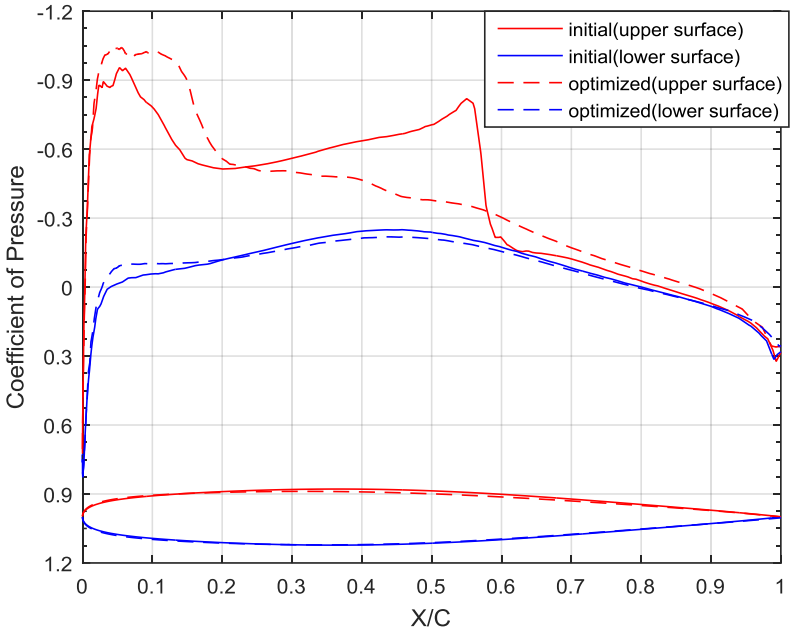


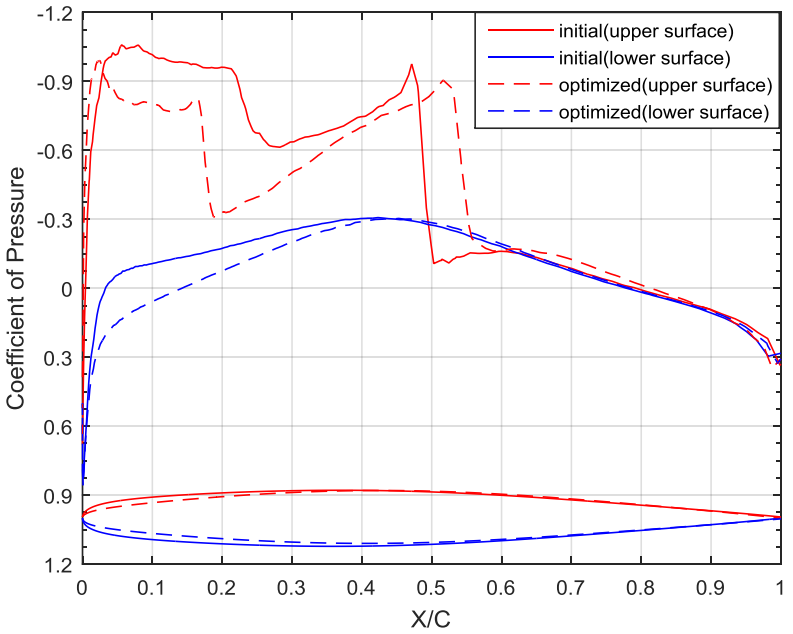
Figure 4.21 Pressure contours for initial and optimized ONERA M6 (with lift constraint)

A reduction in C_d value from 0.011795 to 0.010153 (approximately 14%) is achieved, respecting the lift constraint ($C_l \geq 0.2864$). A comparison between the original and

optimized aerofoil geometries along with the corresponding C_p values is shown in Figure 4.22 and the optimization history in Figure 4.23.



(a)



(b)

Figure 4.22 C_p distribution along the wing at (a) 40% span, (b) 80% span

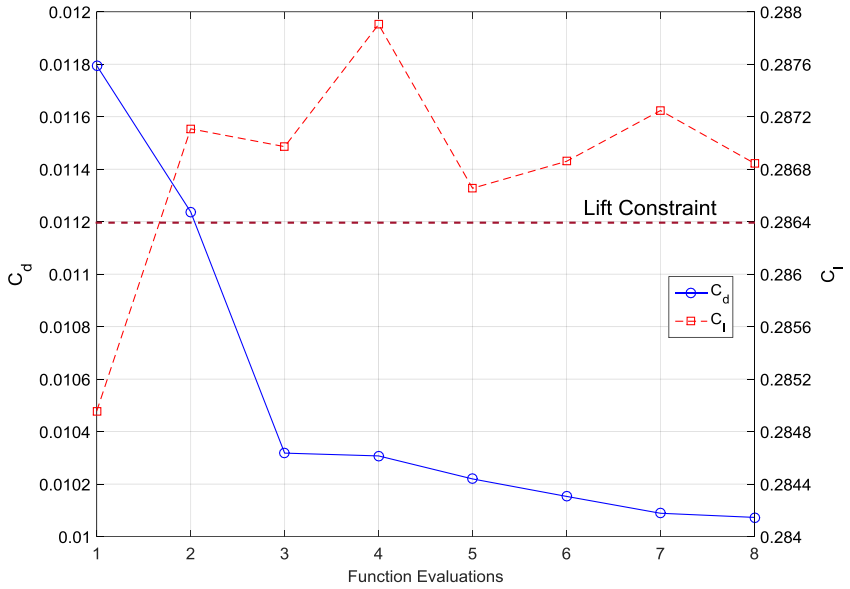


Figure 4.23 Optimization history for minimizing drag on ONERA wing with lift constraint

4.5 NLR 7301 High lift case

To further exercise the proposed methods, a viscous case was examined using the high lift configuration of a NLR7301 aerofoil with trailing edge flap [102]. The flow conditions and optimization problem are defined as:

- Freestream Mach number = 0.185
- Angle of attack (α) = 6^0
- Reynolds Number = 2.51 X 10^6
- Objective Function = max(l/d)
- Turbulence model: Spalart-Allmaras

This test case was particularly chosen to demonstrate the applicability of the developed approaches for high-fidelity aerodynamic optimization. The geometry for the NLR-High lift case was constructed using Bézier curves to define the main wing and the flap. Here, the wing body was considered fixed and only the aerofoil flap was parametrized by considering the positions of Bézier control points defining the upper and lower surface, with a total of 14 CAD parameters to be used in optimization. The CAD model and the feature tree are shown in Figure 4.24.

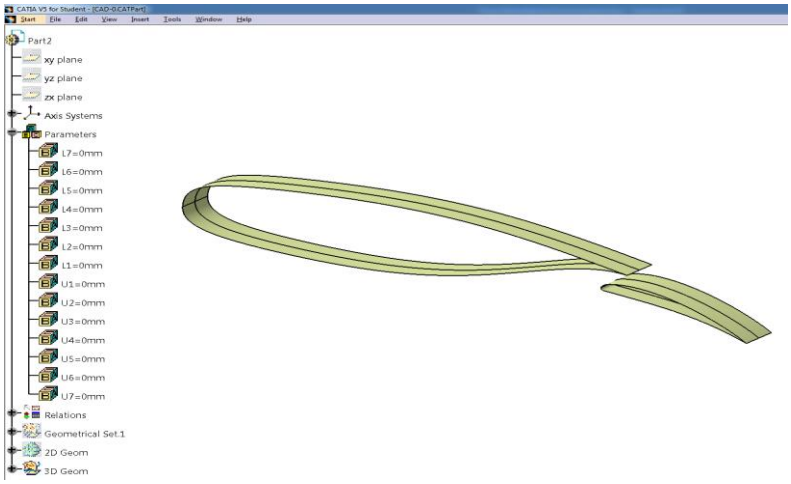


Figure 4.24 CAD model of NLR-7301 high lift configuration

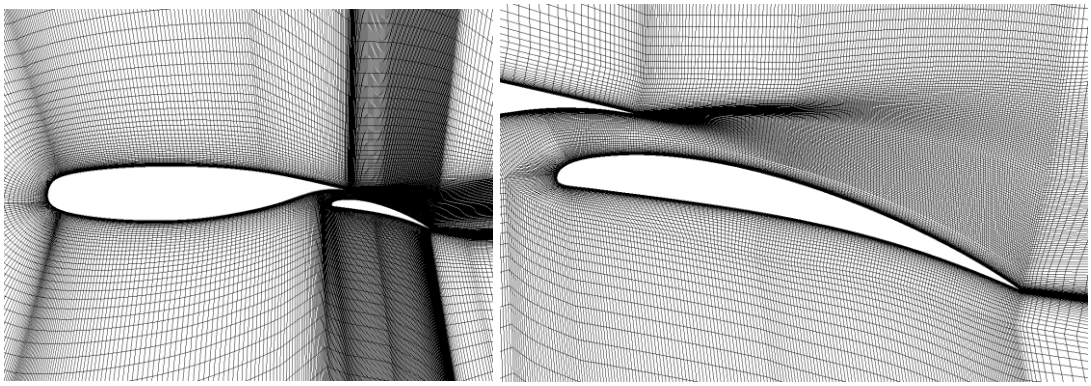


Figure 4.25 Mesh for NLR 7301 wing-body and the flap

The CFD mesh used to start the optimization process was created as a multi-block mesh in ICEM CFD as shown in Figure 4.25, and then exported in CGNS format to be used in SU2. The two-dimensional discretized volume consisted of quadrilateral elements conforming to the aerofoil surfaces. The mesh contained 400 nodes along the wing surface while 420 nodes along the surface of flap. The mesh spacing in the surface normal direction at the aerofoil surface is $1 - 2\ X\ 10^{-6} m$.

In terms of numerical scheme, the convective flux was calculated using the second order Jameson-Schmidt-Turkel scheme with Venkatakrishnan limiter. The turbulence was modelled using Spalart-Allmaras model, and the convective flux was discretized using first-order scalar upwind method. For the purpose of this study, analysis was done using multi-grid approach with a fixed Courant-Friedrichs-Lewy (CFL) number of 4.0, and frozen-turbulence for the adjoint calculations. The convergence of the residuals of the density and corresponding adjoint variable for the initial flap geometry are shown in Figure 4.26. The design velocities for the CAD parameters (Figure 4.27)

were calculated by perturbing each parameter following the methodology outlined in Chapter 3.

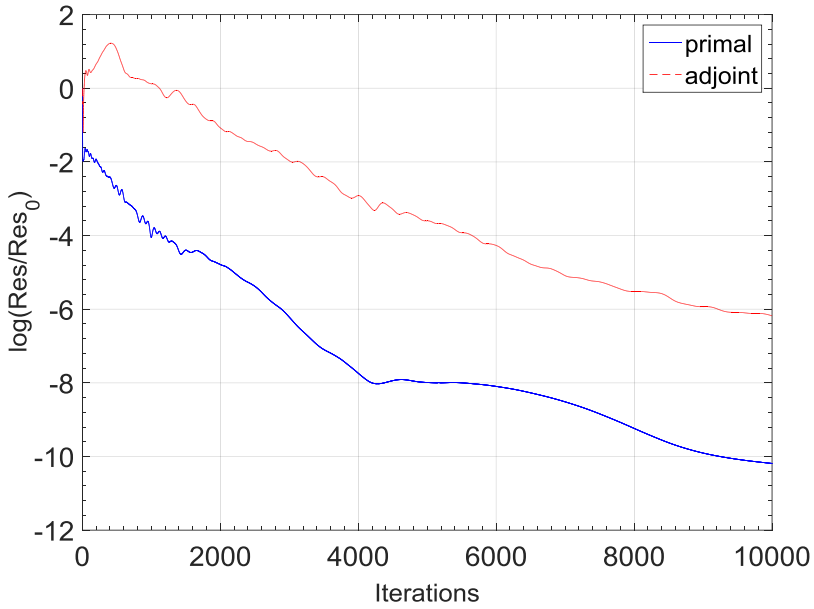
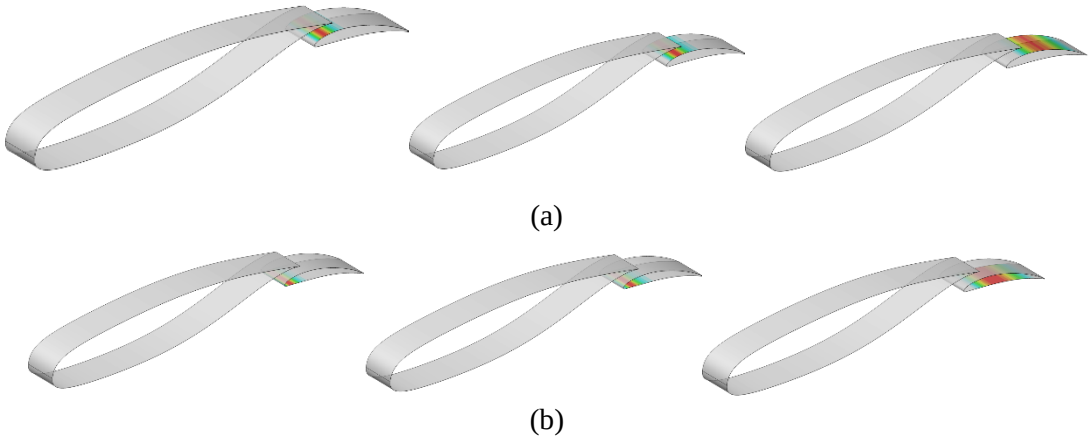


Figure 4.26 Residual convergence for initial NLR flap



(a)

(b)

Figure 4.27 Design velocity contours for CAD parameters on flaps (a) Upper surface (b) Lower surface

The initial and optimized shapes of the flap and the corresponding pressure distribution on the surface is compared in Figure 4.28, while the forces from each element are shown in Table 4.1. The shape optimization of the flap led to a reduction in the gap between the flap and the main wing, causing a larger area of stagnant flow on the trailing edge of the lower surface of the main wing. The impact of the changes on the flap, increases the pressure differential between the surfaces of the main wing, leading to an overall gain in l/d. During optimization, the l/d ratio increased from 79.88 to

81.36 (nearly 1.85%) as shown in Figure 4.29. The pressure distribution on the initial and optimized flap geometry is shown in Figure 4.30.

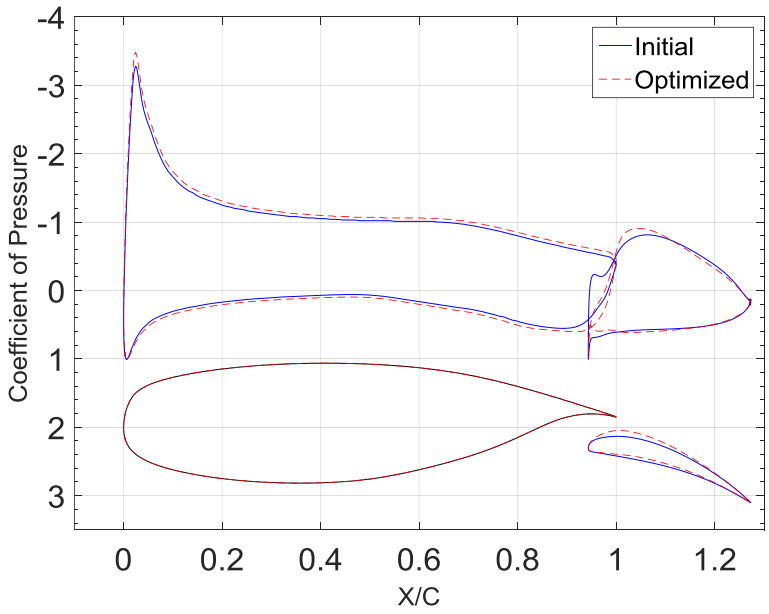


Figure 4.28 C_p and shape comparison of initial and optimized NLR flap

Table 4.1 Force breakdown for initial and optimized NLR geometry

	initial C_l	optimized C_l	initial C_d	optimized C_d
Upper wing	0.8284	0.8785	-0.0508	-0.0581
Lower wing	0.187	0.2268	0.0039	0.0020
Upper flap	0.1186	0.1085	0.0293	0.0396
Lower flap	0.1346	0.1326	0.0334	0.0330

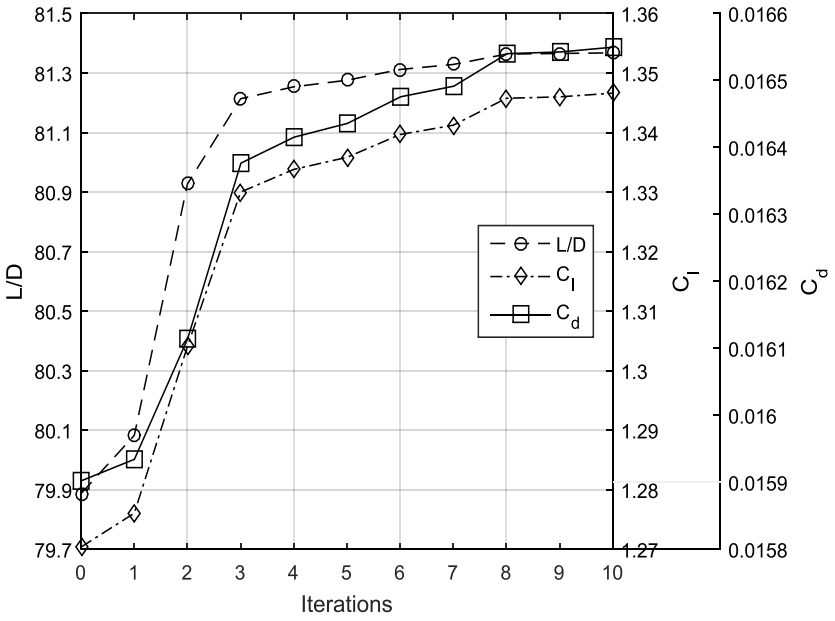
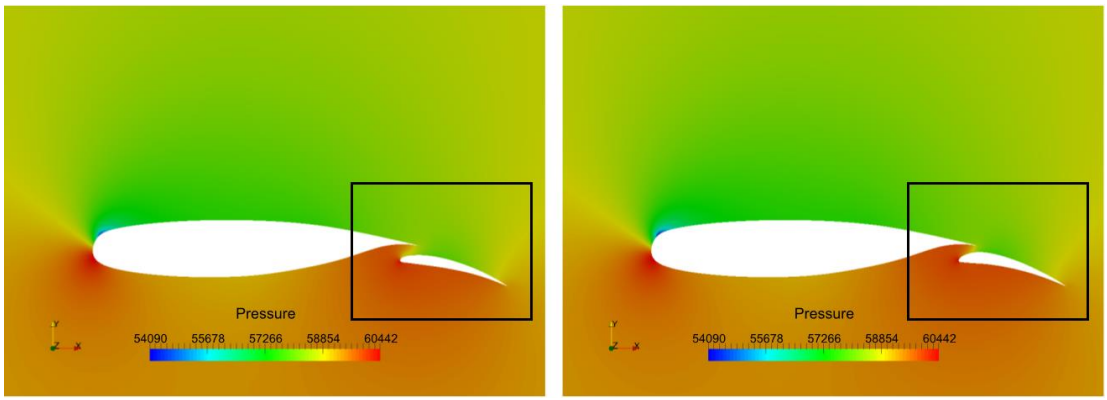


Figure 4.29 Function evaluations during optimization of NLR flap



(a) (b)

Figure 4.30 Pressure contours (a) initial aerofoil (b) optimized aerofoil

4.6 Summary

In this chapter, an automated workflow for CAD-based adjoint optimization framework was presented, which used the methodology developed in Chapter 3 to compute performance gradient with respect to CAD parameters. The applicability and scope of the workflow was demonstrated for optimization of three aerodynamic test cases in inviscid and turbulent flows, and in the presence of geometrical as well as flow constraints. Here, a CAD system API was developed to link CATIA V5 with an optimization framework, which automatically updates the CAD parameter to the values obtained from the optimizer and exports the new CAD model for the primal and adjoint analysis.

Chapter 5 Parametric effectiveness for efficient adjoint optimization

5.1 Introduction

In the preceding chapter, an automated workflow was presented which enables the optimization to be performed directly on feature-based CAD models created in commercial CAD systems. The benefit of using this approach is that assuming the original CAD model was well created, the constraints on shape imposed by the features in the CAD model feature tree would mean that the optimized part can be manufactured, providing the features were well chosen. To a large extent this will depend on the skill and experience of the CAD model creator, and their ability to visualize and parameterize the design space. The downside of using a feature-based CAD model to optimize the design is that the parameters are not primarily chosen with optimization in mind, and often it is not obvious from the parameterization which parameter value(s) need to be modified to achieve the desired shape change, especially when the person implementing the change is not the creator of the CAD model. Robinson et al. [103] used adjoint sensitivities and design velocities to define the measure of parametric effectiveness to rate the quality of a CAD parameterizations to be used for optimization. Parametric effectiveness compares the maximum change in performance that can be achieved using the parameterization, to the maximum performance improvement that could be obtained if the model is free to move (i.e. not constrained by any parameterization), where both are subjected to the constraint of a unit root-mean-squared boundary movement.

The aim of this chapter is to present an automated approach to efficiently calculate the parametric effectiveness for any set of parameters defined within the CAD modelling system CATIA V5. In this work, the approach is used to automatically select a subset of parameters which provides the greatest potential for performance improvement in an acceptable time-scale. The ability to down-select the most effective set of

parameters is advantageous because, while one of the benefits of adjoint optimization is that the cost of calculating sensitivities is virtually independent of the number of design variables, the cost of modifying the CAD model of industrial complexity by changing all parameters during an optimization step is potentially high.

5.2 Computing parametric effectiveness

While the original work of Robinson et. al. [103] proposed the measure and was able to compute it for all parameters, it was unable to suggest an automated approach which could be used to compute the parametric effectiveness of subset of parameters, or to identify the most effective set of parameters in industrially acceptable time-scales. When computing parametric effectiveness, a constraint on overall boundary movement is imposed for each parametric perturbation. This ensures that a parameter moving an area of low sensitivity by a large amount would not be favoured over the parameters causing a small localised movement in the areas of high sensitivity. Parametric effectiveness ranges from 0 to 1. A high value of parametric effectiveness indicates that the parameters in the model can cause the shape to change in exactly the manner that the adjoint sensitivity map suggests. As the value of parameters approach their optimum values during the optimization, the parametric effectiveness tends to zero. The detailed mathematical derivation is provided in [103], and a short description is given here.

Consider, the root-mean square design velocity over the boundary is dV as

$$dV = \sqrt{\frac{\int_A V_n^2 dA}{A}} \tag{5.1}$$

The boundary movement constraint can be included in the formulation of objective function using Lagrangian formulation. Let the constraint be represented by I_1 in

$$I_1 = \int_A \frac{V_n^2 dA}{A dV^2} = 1 \tag{5.2}$$

The Lagrangian for the constrained system can be written as

$$L = dJ + \lambda_1(I_1 - 1) = -\int \phi V_n dA + \lambda_1 \left(\int_A \frac{V_n^2 dA}{A dV^2} - 1 \right) \tag{5.3}$$

The maxima and minima of Eqn 5.3 occurs when

$$\frac{\partial L}{\partial \lambda_1} = 0; \frac{\partial L}{\partial V_n} = 0 \tag{5.4}$$

Eqn. 5.3 can be differentiated with respect to λ_1 and V_n as

$$\frac{\partial L}{\partial \lambda_1} = \int_A \frac{V_n^2 dA}{AdV^2} - 1 = 0. \tag{5.5}$$

$$\frac{\partial L}{\partial V_n} = -\int \phi dA + \frac{2\lambda_1}{AdV^2} \int_A V_n dA = 0. \tag{5.6}$$

The solution of Eqn. 5.6 is the optimum design velocity V_n^*

$$V_n^* = \frac{AdV^2}{2\lambda_1} \phi. \tag{5.7}$$

Using Eqn. 5.7 in Eqn. 5.5 to get the value of λ_1 as

$$\lambda_1 = \frac{AdV^2}{2} \sqrt{\int_A \frac{\phi^2 dA}{AdV^2}}. \tag{5.8}$$

Using value of λ_1 in Eqn. 5.7, the optimum design velocity can be computed as

$$V_n^* = \frac{1}{\sqrt{\int_A \frac{\phi^2 dA}{AdV^2}}} \phi \tag{5.9}$$

The optimum change in performance for an un-constrained model can be obtained as

$$\left(\frac{dJ}{dV}\right)_{optimum} = -\sqrt{A \int_A \phi^2 dA}. \tag{5.10}$$

In the case where boundary movement is caused by changing a CAD parameter, and by assuming the optimum parametric performance improvement is obtained by perturbing the parameters in the direction of steepest decent, the vector of parameter changes can be written as

$$dP = k\{S_1 S_2 \dots \dots\}, \tag{5.11}$$

where k is a multiplier specifying the magnitude of the steepest decent vector. The resulting design velocity due to parameter change can be defined as

$$V_n = kS_1 V_{n1} + kS_2 V_{n2} + \cdots \tag{5.12}$$

The above equations can be combined to obtain the performance change per unit of root-mean-square design velocity, specified by the parametric change as

$$\left(\frac{dJ}{dV}\right)_{param} = -\sqrt{\frac{A}{\int_A (\sum_{i=1}^n S_i V_{ni})^2 dA}} \sum_{i=1}^n (S_i)^2, \tag{5.13}$$

In this work, an automated method to compute Eqn. 5.10 and Eqn. 5.13 is developed, which allows the computation of parametric effectiveness for any combination of CAD parameters. It has been implemented in Python 3.5 for design velocities

computed from CATIA V5 and SIEMENS NX, and adjoint sensitivity maps from HELYX.

5.3 Automated approach for CAD parameter selection

It is shown in [103] that parameters selected based on parametric effectiveness are potentially better at localising the shape change in regions of high adjoint sensitivities. It also states that the most effective set of parameters may not include all parameters and should be identified using a power-set approach. This requires the parametric effectiveness to be calculated for all possible combinations of parameters. While this could be achieved in a brute-force manner i.e. by evaluating Eqn 5.13 for all the possible combinations of parameters by formulating a power-set. The power-set of any set $\mathbb{Q}$ of n parameters is the set of all subsets of $\mathbb{Q}$ (including the empty set) giving a total of $2^n - 1$ different parametric combinations. The implementation of the power-set approach is therefore computationally prohibitive when the number of parameters is large, as it is for most industrially relevant CAD models. Also, to reduce the cost of industrial optimization Design of Experiments based methodologies are popular, e.g. [104]. These are used to either screen out parameters predated to have a small influence on performance, and/or to generate response surfaces which are then used for optimization. The issue with these approaches are that they require many function evaluations to obtain sufficient data to formulate the process.

In this chapter an approach is formulated to efficiently obtain the optimum parametric combination giving highest parametric effectiveness without exhaustively evaluating Eqn. 5.13 for all parametric combinations. This is achieved by:

Step 1: It is assumed that parameters with very low individual parametric effectiveness can be neglected, therefore all parameters with an individual parametric effectiveness greater than 0.02 are selected. The number of parameters $= n$.

Step 2: for the n parameters in Step 1, all the possible combinations of 2 parameters (the sets are ordered based on label) are created each referred as a **set**. Here, $\mathbf{C}_2^n$ sets are formed, where $\mathbf{C}$ is a combinatorial operator.

Step 3: The parameter sets are now grouped together such that $n - 1$ **groups** are created to contain parameter sets with the same first parameter.

Step 4: The parametric effectiveness of each set in a group is computed, and the set with highest parametric effectiveness is selected for that group.

Step 5: The selected set from each group in Step 4, is then combined with all the remaining $n-2$ parameters and analyzed to get a new set of parameters with higher parametric effectiveness.

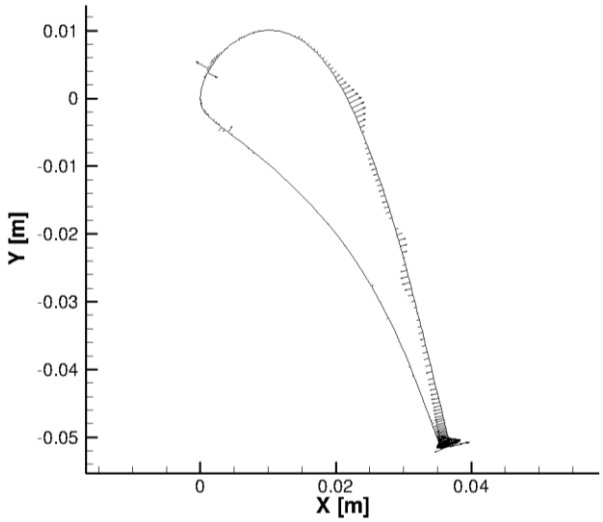
Step 6: If the resulting parametric effectiveness in step 5 is less than that obtained in step 4, then the highest achievable parametric effectiveness for that group is the one obtained in Step 4. Else, Step 4 to Step 6 are repeated.

Step 7: when adding a new parameter does not cause the parametric effectiveness of a set to increase, then the maximum value among the groups is sought. The parameter set with maximum parametric effectiveness is identified as the subset of all parameters with the highest parametric effectiveness.

5.4 Example applications

5.4.1 LS89 test cases

The LS89 test case was previously analyzed in Chapter 3, where the details of the CAD model were shown in Figure 3.16. Here, the LS89 geometry was modelled using 32 design parameters as shown in Table 5.1. Please note that the trailing edge radius and axial chord length of the model were fixed to meet the manufacturing requirements. Here the objective was to minimize the entropy generation, and the adjoint analysis was done using VKI in-house code. The adjoint surface sensitivities for LS89 are shown in Figure 5.1.



[4]Figure 5.1 Adjoint sensitivity for LS89

[4] Courtesy: Ismael S. Torreguitart, Von Karmann Institute of Fluid Dynamics, Belgium

In order to demonstrate the scope of parametric effectiveness to rate the quality of different combination of CAD parameters, the current LS89 parameterization was sub-divided into four other categories as outlined in Table 5.2. The parametric effectiveness computed for each parametric combination showed that the SS parameters were much more efficient in reducing the objective function for the same overall boundary movement (ΔV), while the effectiveness was reduced when the Camber line knot positions were used in the parameterization scheme. Similar results were obtained for reduction in objective function (ΔJ) when CFD analysis was performed on geometries perturbed to achieve the same (ΔV) as outlined in Table 5.3.

Table 5.1 LS89 parameters

original parameterization	symbol
trailing edge thickness	D_{TE}
leading edge radius	R_{LE}
inlet angle	β_{in}
outlet angle	β_{out}
PS wedge angle	φ_{PS}
SS wedge angle	φ_{SS}
PS thickness	$t_{PS}^1, t_{PS}^2, t_{PS}^3, t_{PS}^4$
SS thickness	$t_{SS}^1, t_{SS}^2, \ldots., t_{SS}^9$
Camber line knot parameters for PS	$K_{PS}^1, K_{PS}^2, K_{PS}^3, K_{PS}^4$
Camber line knot parameters for SS	$K_{SS}^1, K_{SS}^2, \ldots, K_{SS}^9$

Table 5.2 Four different parametric combinations of LS89

Parameterization-1	Parameterization-2	Parameterization-3	Parameterization-4
D_{TE}	D_{TE}	D_{TE}	D_{TE}
R_{LE}	R_{LE}	R_{LE}	R_{LE}
β_{in}	β_{in}	β_{in}	β_{in}
β_{out}	β_{out}	β_{out}	β_{out}
φ_{PS}	φ_{PS}	φ_{PS}	φ_{PS}
φ_{SS}	φ_{SS}	φ_{SS}	φ_{SS}
$t_{PS}^1, t_{PS}^2, t_{PS}^3, t_{PS}^4$	-	$t_{PS}^1, t_{PS}^2, t_{PS}^3, t_{PS}^4$	-
$t_{SS}^1, t_{SS}^2, \ldots., t_{SS}^9$	$t_{SS}^1, t_{SS}^2, \ldots., t_{SS}^9$	-	-

Table 5.3 Parametric Effectiveness & objective function change for LS89 parameters

	Parametric Effectiveness	ΔJ	ΔV
Original parameters	0.030	0.096%	2.5×10^{-5}
Parameterization-1	0.0921	0.31%	2.5×10^{-5}
Parameterization-2	0.0948	0.33%	2.5×10^{-5}
Parameterization-3	0.0046	0.0035%	2.5×10^{-5}
Parameterization-4	0.008	0.0155%	2.5×10^{-5}

The motivation for selecting LS89 test case is that it is simple geometry and thus easy to manually relate the effect of different parameters. Analyzing the adjoint sensitivity map, it was inferred that the pressure side of the blade has negligible contribution towards minimizing the objective. The parametric effectiveness calculation also revealed the same information. Further, the CFD analysis showed the same results and is interesting to observe a one-to-one correlation between the parametric effectiveness and the change in objective function.

5.4.2 Automotive ventilation duct

A 3D parametric CAD model of an automotive ventilation duct was obtained from the Volkswagen Group Research (VW) and is shown in Figure 5.2. The details of the adjoint analysis are described in [16], and the adjoint sensitivity map is shown in Figure 5.3, where blue regions must be pushed inside the body (towards the fluid) while red are to be pulled outside the body (away from the fluid) to reduce the objective function. The CAD model was created in CATIA V5 and had 263 real-valued parameters that when perturbed change the shape of the duct. Here, the parametric effectiveness of two different sets of parameters was considered. The first set was a combination of four parameters that were selected by the designer to perturb the model mostly in the areas of high adjoint sensitivity, as detailed in the original paper [103]. This selection of parameters was based on the engineering judgement of the designer and their identification required considerable amount of time and skill to select. The other parametric set was that obtained using the approach presented in section 5.3.

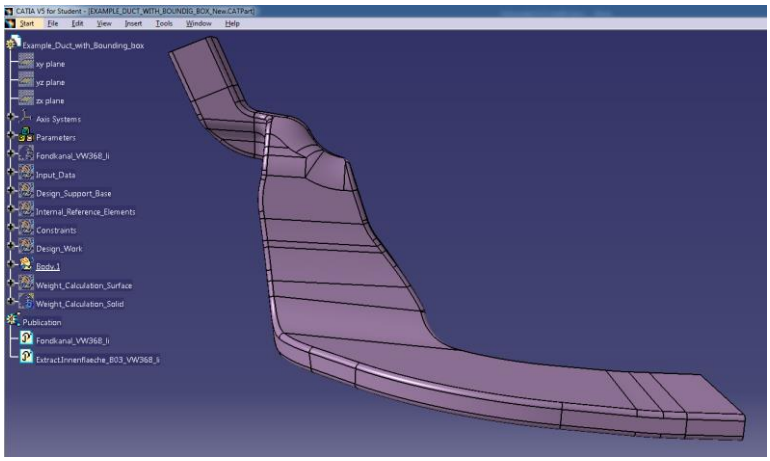


Figure 5.2 Parametric CAD model of an automotive duct

The parametric effectiveness of the parameters selected by designer was 0.47 compared to 0.53 for the most effective parametric combination (which consisted of 16 parameters). Interestingly, the parameters obtained using the developed approach also contained the four parameters that were selected by the designer. Figure 5.4 shows the contours of design velocity when the designer's parameters and the most effective parametric combinations were perturbed to move the boundary in the steepest decent direction. In both cases, the overall boundary movement caused by the change is the same and is constrained to be small ($dV = 1E^{-4}$). Figure 5.4(c) shows that the design velocity for the most effective parametric combinations move the model boundary in a manner close to that suggested by the adjoint sensitivity map in Figure 5.3, giving confidence in the applicability of the developed process.

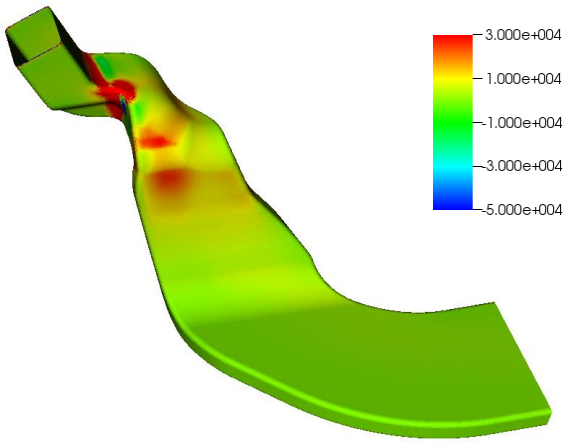
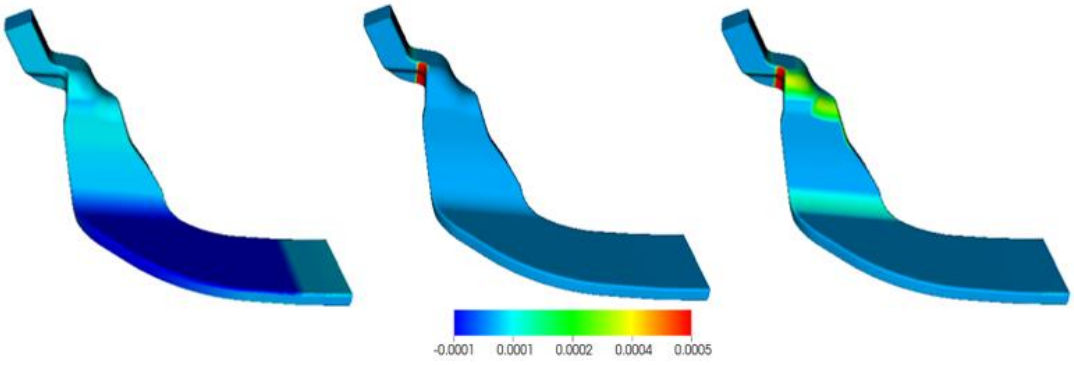


Figure 5.3 Adjoint sensitivity map of automotive duct



(a) (b) (c)

Figure 5.4 Design velocity for the same overall boundary movement $dV = 1\ E^{-4}$, (a) all parameters, (b) designer's parameters, (b) most effective parameteric combination

5.4.3 S-Bend Duct

In the next test case, a parametric CAD model of the S-Bend duct [29] is created in CATIA V5 as shown in Figure 5.5. It was modelled using eight 2D sketches at

different positions and orientations along the length of the duct, and then developing a multi-section solid passing through these sketch profiles. The duct is composed of three individual sections i.e. inlet, S-Bend and outlet as shown in Figure 5.5. As the inlet and outlet ducts will join with other components their shape is fixed, so they are not considered for optimization. Here the optimization variables are the parameters defining the four sketches (shown in broken lines) describing the interior profile of the S-Bend (48 parameters).

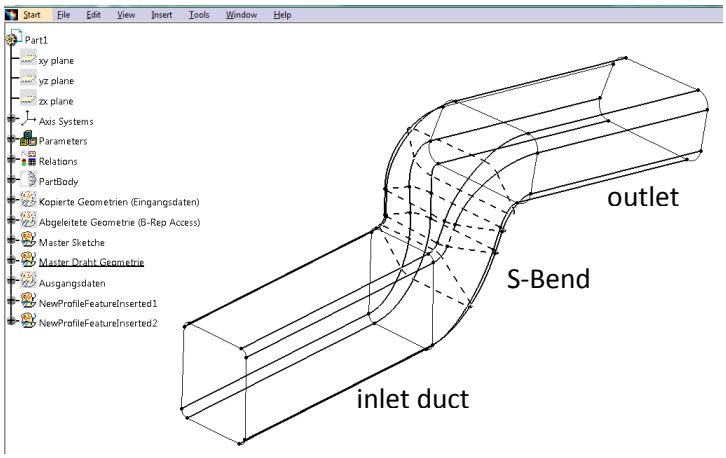


Figure 5.5 CAD model of S-Bend duct

The objective function considered for optimization is the power dissipation through the duct [16], defined as

$$J = \int_A v \cdot \hat{n} \left(p + \frac{1}{2} v^2 \right) dA \,, \tag{5.14}$$

where p and v are the pressure and velocity of the flow, and A and $\hat{n}$ are the surface of the duct and its unit normal, pointing away from the fluid area. The flow is laminar, with Reynolds number $R_e = 350$, calculated with a hydraulic diameter of $D_h = 0.053m$, inlet velocity $u = 0.1m/s$ and kinematic viscosity $v = 1.511\ X\ 10^{-5}\ m^2/s$. The computational mesh is created in ICEM-CFD [98] with approximately 250,000 elements. The flow equations are solved using the standard steady state incompressible OpenFOAM© solver simpleFoam. The adjoint equations are solved using the Helyx adjoint solver, provided by ENGYS [105]. The adjoint sensitivity sensitivities of the objective function calculated using the continuous adjoint approach is shown in Figure 5.6, where the surface is to be pulled outside the body (away from fluid) in red regions and to be pushed inside the body (towards the fluid) in the blue regions.

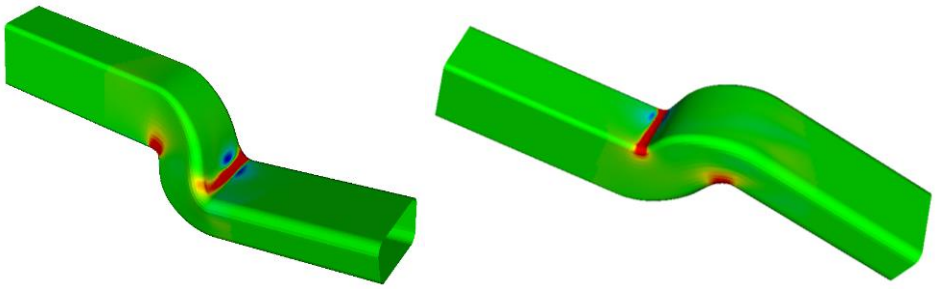


Figure 5.6 Adjoint sensitivities map: to minimize the objective function the surface should be pulled out at red regions and pushed in at blue regions

Here the approach presented in section 5.3 is used to identify a subset of the full parameter set with the highest parametric effectiveness. In this test case, the most effective parameter set contained 13 parameters and had an effectiveness of 0.66 compared to an effectiveness of 0.60 obtained for all 48 parameters.

Table 5.4 Time required to update S-Bend CAD model

	Original CAD model (48 parameters)	Most effective parameter (13 parameters)
time for one CAD update	32 s	8 s

The computational effort required to update the S-Bend model in CATIA V5 using the different parameter sets is shown in Table 5.4, where it can be seen that updating all the parameters of the CAD model is comparatively more expensive compared to updating a selected set of parameters (factor of 4). In this case, the time required for CAD updates is still small compared to the flow and adjoint solution time (45 minutes).

Optimization

The shape of the S-Bend duct was optimized using the SLSQP method implemented in Scipy [94]. For the S-Bend duct the optimization is performed using two different sets of parameters, firstly using all of the CAD parameters and secondly using the subset of parameters with the highest parametric effectiveness. At each optimization step, a new computational mesh is created in ICEM-CFD using an automated process.

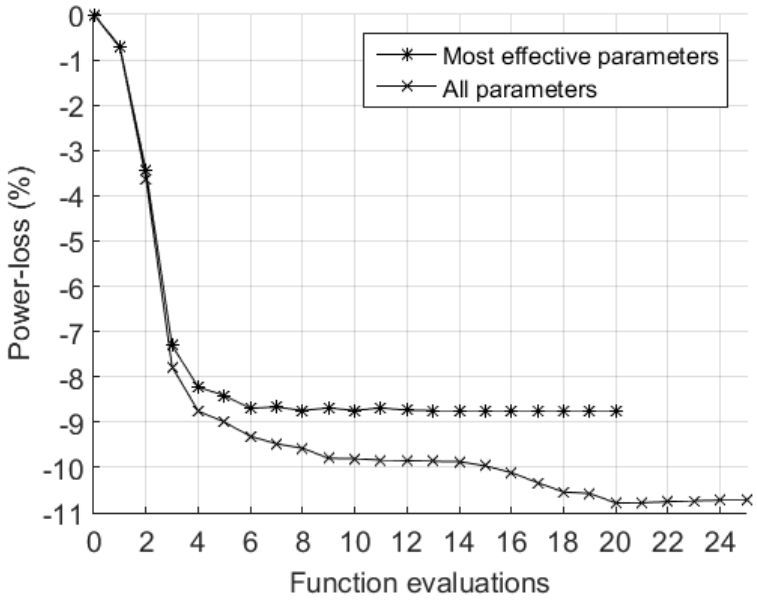


Figure 5.7 Change in objective function during S-Bend optimization

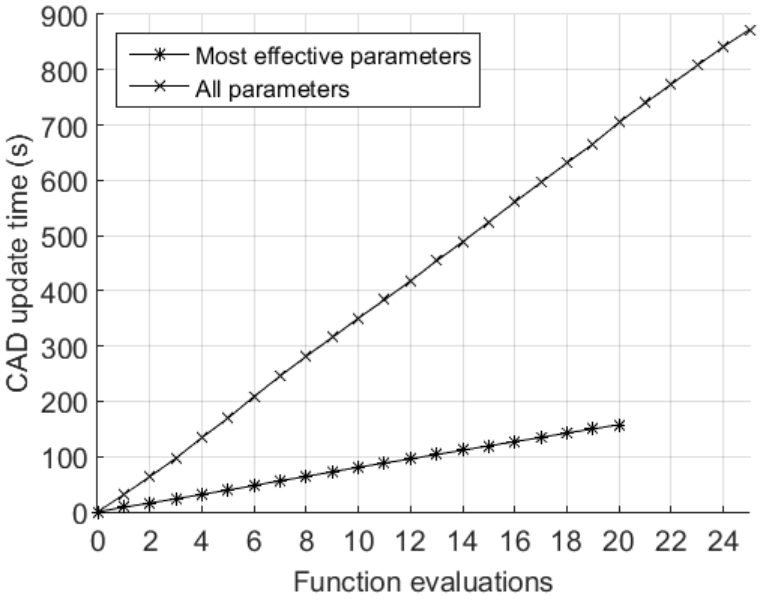


Figure 5.8 Time taken to update S-Bend CAD model during optimization

The optimization history for minimizing the power-loss across the duct is shown in Figure 5.7. A reduction in power-loss by 10.72 % is observed when all the parameters are used for optimization, compared to 8.75 % when the parameters with highest parametric effectiveness are used. The total time required to update a parametric CAD model (Figure 5.8), is higher when all parameters are used to update the CAD model compared to updating a small set of parameters. It should be remembered that a CAD update is required for each iteration during the optimization, and that this step of CAD updating cannot be parallelized.

Table 5.5 shows the total time taken by SLSQP optimization for two different sets of parameters. It can be seen that even for this relatively simple test case with a reasonable smaller number of design parameters, reducing the number of parameters for optimization resulted in a time saving of approximately 9,000 seconds (a 29% reduction). The reduction in the gain in performance was approximately 2% (Figure 5.7).

Table 5.5: Time statistics for S-Bend optimization

	Original CAD (48 parameters)	Most effective parameter (13 parameters)
total time taken for updating CAD during the optimization	873 s	165 s
total time for optimization	31360 s	22375 s

5.4.4 TUM DrivAer mirror

Another test case was formulated in collaboration with the Volkswagen Group Research with the objective of reducing automotive noise with the use of a surrogate model for aeroacoustics [106]. Here the objective function was to minimize noise as perceived by a driver of the car. The model under investigation is the TUM DrivAer vehicle [107], using a fast-back configuration with smooth underbody and closed wheels. The CAD model of the car mirror was initially provided as a STEP file. A replica model was created by Mr. Gary Stevens (BEng student, QUB) using CATIA V5 using a series of points, lines and splines. The wireframe of the CAD model is shown in Figure 5.9. The surface fitting methods in CATIA V5 (like multi-section surface and fill surface) were used to create the outer surfaces and produce a 3D CAD model of the mirror with 2925 CAD parameters.

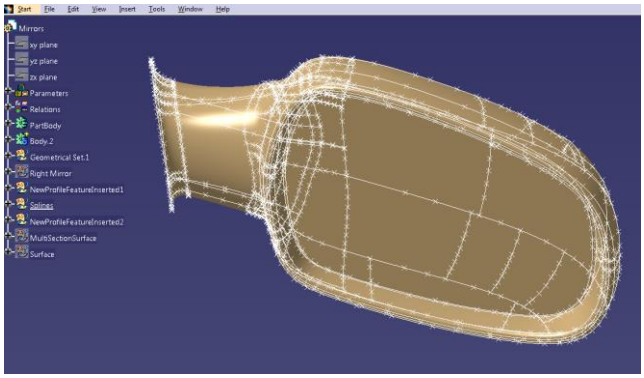


Figure 5.9 Parametric CAD model of car mirror (wireframe)

The parameters were the x, y and z position of the points in Figure 5.9 and the resulting CAD model is shown in Figure 5.10. Each parameter controlled the position of

individual points. For the flow and adjoint analysis, half of the car was meshed. The computational grid consisted of 5 million cells. As seen in Figure 5.11, a grid refinement is used around the mirror for a better turbulence resolution in this area.

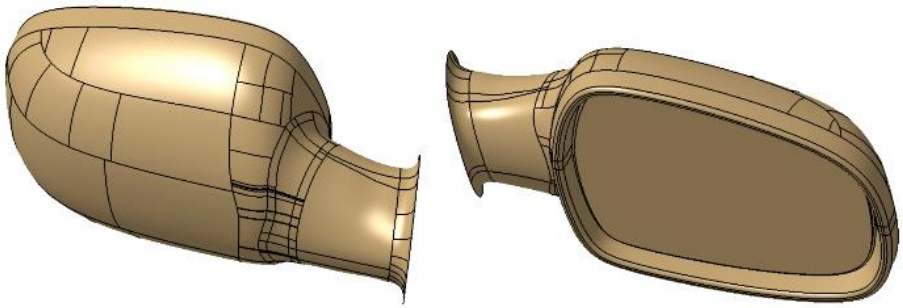
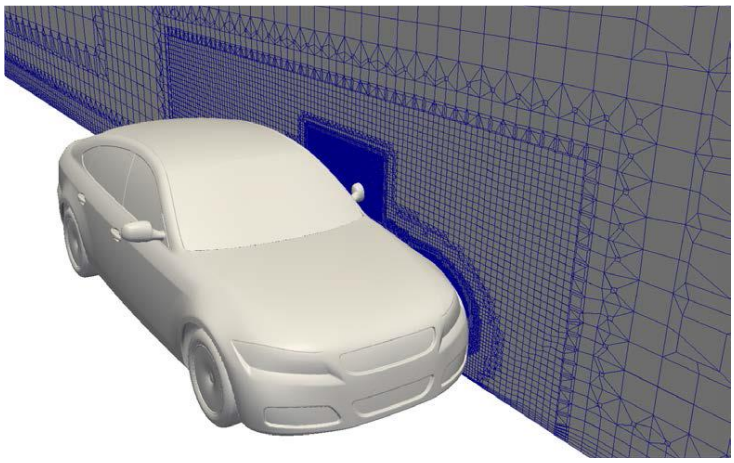


Figure 5.10 CAD model of car mirror (solid)



[5]Figure 5.11 Slice of the computational grid around the DrivAer vehicle. The refinement boxes around the car and the mirror can be seen

The flow equations were solved using the standard steady state incompressible OpenFOAM© solver simpleFoam. The adjoint equations were solved using the HELYX Adjoint solver, provided by ENGYS [105].

The turbulence model used is given by

$$R_i^{\tilde{\nu}} = v_j \frac{\partial \tilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\tilde{\nu}}{\sigma}\right)\frac{\partial \tilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial x_j}\right)^2 - \tilde{\nu}P + \tilde{\nu}D = 0, \qquad (5.15)$$

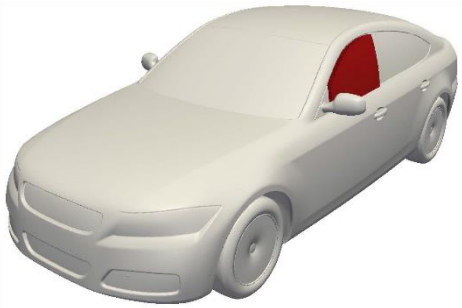
where v_i is the flow velocity, ν is the kinematic viscosity, $\tilde{\nu}$ is the turbulence state variable, P and D are the turbulence production and dissipation terms respectively.

The eddy viscosity coefficient is expressed as

[5] Courtesy: Christos Kapellos, Volkswagen Research, Germany

$$\nu_t = \tilde{\nu} f_{v1}(\tilde{\nu}), \tag{5.16}$$

where f_{v1} is the function obtained from the turbulence model. For the treatment of ν_i and ν_t on the wall boundary faces, the wall function technique was employed. The low frequency noise perceived inside the cabin can be linked to the turbulence level at the area directly outside of the driver side window as shown in Figure 5.12.



[6]Figure 5.12 Volume (in red) over which the objective function is integrated. The volume was created by the extrusion of the DrivAer driver window by 3 cm

In this sense, a surrogate aeroacoustics objective function was formulated as the integral of the squared turbulent viscosity over a volume near the side window as

$$F_{noise} = \int_{\Omega} \nu_t^2 \, d\Omega. \tag{5.17}$$

The objective function was augmented by the scalar product of the state equation residuals with the adjoint variables, yielding

$$J = F + \int_{\Omega} u_i R_i^v \, d\Omega + \int_{\Omega} q R^p \, d\Omega + \int_{\Omega} \tilde{v}_a R^{\tilde{\nu}} \, d\Omega. \tag{5.18}$$

It is important to note that without the differentiation of the turbulence model, relying on the "frozen turbulence" assumption, dealing with an optimization problem of this kind would not be possible. This is because the objective function itself depends on the turbulent variable $\tilde{\nu}$. Herein, to formulate the continuous adjoint method the fully differentiated Spalart-Allmaras turbulence model based on wall functions was used [108]. The adjoint sensitivity map is computed during the first optimization cycle and is presented in Figure 5.13, where red areas must be pushed inside the body (away from fluid) while blue are to be pulled outside the body (towards the fluid) to reduce the objective function.

[6] Courtesy: Christos Kapellos, Volkswagen Research, Germany

[7]Figure 5.13 Adjoint sensitivity maps targeting at turbulent noise minimisation, seen from top (left) and bottom (right)

Here the parametric effectiveness approach was used to identify the subset of full parameter set (2925 parameters) with the highest parametric effectiveness. In this case, the most effective parameter set contained 48 parameters with a parametric effectiveness of 0.83. The computational effort required to update the parametric DrivAer model is shown in Table 5.6. It can be seen that updating all the parameters of the CAD model is computationally much more expensive.

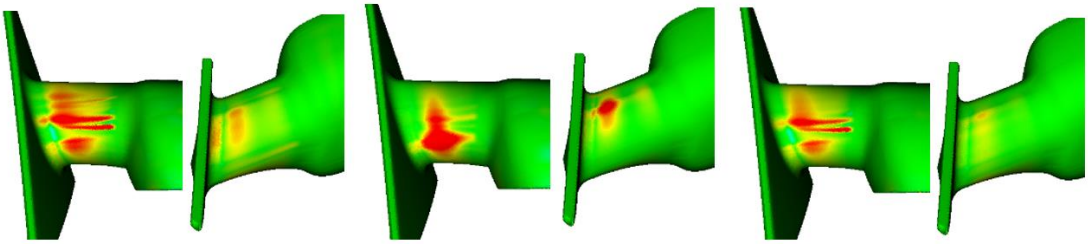
Table 5.6 Time required to update the DrivAer CAD model

	Original CAD model (2925 parameters)	Most efficient parameter (48 parameters)
CAD update time	10716 s	129 s

An alternative approach for reducing the number of parameters would be to select those with the highest parametric sensitivity. The benefit of using parametric effectiveness to select the set of parameters for optimization was further substantiated by comparing the design velocities when the model is perturbed using all the 2925 parameters (Figure 5.14(a)), the most effective parametric combination (consisting of 48 parameters) found using the approach presented in this chapter (Figure 5.14(b)), and the same number of parameters with the highest parametric sensitivities (Figure 5.14(c)). In all cases, the parameters are perturbed in the steepest decent direction and the overall boundary movement for the perturbations is kept small ($dV = 3E^{-5}$). Comparing Figure 5.13 and Figure 5.14, it is seen that the parametric combination with highest parametric effectiveness moves the model such that the boundary displacement

[7] Courtesy: Christos Kapellos, Volkswagen Research, Germany

is highly focussed in the areas of high sensitivity and very little in other regions, while other parametric combinations move the boundary of the model in a less focussed fashion.



(a) (b) (c)

Figure 5.14 Design velocity for the overall boundary movement $dV = 3\ E^{-5}$, (a) all 2925 CAD parameters, (b) most effective parameteric combination (48 parameters), (c) 48 parameters with highest sensitivity

Optimization

The shape optimization of the car mirror was performed using a steepest descent strategy. For each optimization step, the design velocity method described in Chapter 3 was used to deform the surface mesh points, and a linear elasticity approach was used to deform the volume mesh [109]. After the optimization algorithm converged, the optimal geometry was 6.8% "quieter" when using the most effective parametric combination, compared to 4.1% when the 48 parameters with highest parametric sensitivity were used. Comparing the design velocity of the optimized designs for the initial and optimized geometry obtained using most effective parameters (Figure 5.15), it is seen that the top and bottom of the neck of the mirror has been pushed in to suppress the generation of turbulence on the wake of the mirror, consequently reducing the turbulence viscosity flowing through the volume over which the objective function is integrated (Figure 5.16).

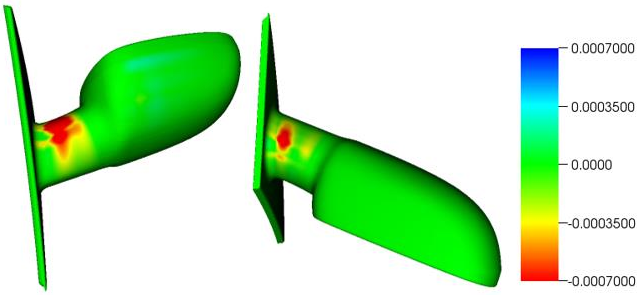
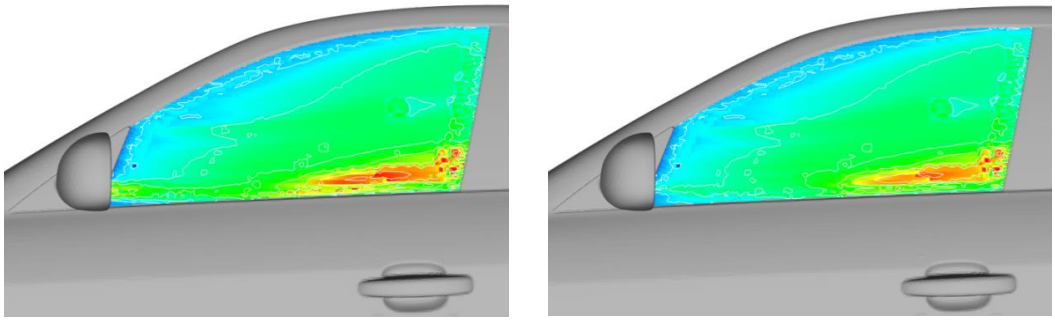


Figure 5.15 Design velocity contours for the CAD model optimized using the most effective parametric combination



[8]Figure 5.16 Squared turbulent viscosity computed at a slice of the volume over which the objective function is integrated. (original shape (left), optimized using 48 most effective parameters (right))

5.5 Summary

The main objective of this Chapter was to present an automated approach to compute the parametric effectiveness of the parameters defined within a CAD modelling system, and subsequently use the information to select the optimum combination of CAD parameters to be used for optimization. The developed approach was demonstrated on four different test cases. Firstly, it was shown that parametric effectiveness can be used as an effective measure to rate the quality of different parameterization strategies. The claims were strengthened by performing the CFD analysis to quantify the change in objective function when CAD models were perturbed to obtain the same boundary movement. Subsequently, it was shown that the parameters selected based on parametric effectiveness were better in localizing the boundary movement in the areas of high adjoint sensitivities. Thereafter, the optimization results were shown for two test cases, comparing the performance gained by using the complete design space with its subset obtained using the approach developed in this work. It was shown that even for the optimization of a simple test case (S-Bend duct) with a smaller number of design parameters, reducing the number of design parameters resulted in a time saving of 29%, at the cost of reduced gain in performance by approximately 2%. Later, it was shown for the DrivAer model that as the complexity of CAD model increases in terms of the number of design variables, the time saving achieved by reducing the design space increases substantially.

[8] Courtesy: Christos Kapellos, Volkswagen Research, Germany

Chapter 6 Automatic refinement of CAD parameterization

6.1 Introduction

In preceding chapters, it was shown that the CAD model created in a feature-based CAD modelling system can be directly optimized following a gradient based approach using design velocities and adjoint sensitivities. The success of the optimization frameworks depends to a large extent on the skill and experience of the CAD model creator, and their ability to visualize and parameterize the design space. Sometimes, this could result in situations where the design may not be the best choice for optimization, and it becomes essential to either re-parameterize the existing features or add more features to increase the flexibility of the model, if an optimum result is to be achieved. This is not always straight-forward, especially if the person implementing these modifications is not the initial creator of the CAD model and is not fully aware of how it was constructed. In such cases, understanding the response of the model to a change in features or parameters is difficult to determine a priori.

In this chapter, a novel automated methodology is presented which increases the design flexibility by automatically refining the existing or inserting new shape features into the CAD model. New features are inserted directly into the CAD model feature tree, and so the model is a valid CAD representation and can be easily used for down-stream applications, without any additional post-processing relative to a standard model. The design decision on which CAD features to insert, and where to place it, is made by analyzing the adjoint sensitivities.

Here, a new route to optimization is sought, where the parameterization of the model is modified when it is determined the current parameters are not fit for the purpose and the full set of feature parameters (new and original) are used to change the shape of the model. In this work, this is achieved by both updating the sketch entities i.e. straight lines to splines (with additional control points), and by adding new sketch profiles to

modify the definition of a multi-section solid feature that is common in both aerospace and automotive designs. This has been implemented in the commercial CAD modelling system CATIA V5. The automated modules for inserting different CAD features are developed in Python 3.5 where an interface with CATIA V5 was created using win32com object programming [110].

6.2 Methodology for feature insertion

In Chapter 5, a methodology was presented to compute the parametric effectiveness of CAD model parameters, where a low value (~ 0) indicated that the parameters in the model were not able to perturb the CAD model shape in a manner close to which the adjoint sensitivity map suggests, while a higher value (~ 1) indicated that there was a set of parameters that could perturb the boundary close to the manner suggested by the adjoint sensitivity. For a CAD model which is found to have a low parametric effectiveness, this chapter describes four approaches to increase its parametric effectiveness through adding new features and associated parameters to the CAD model feature tree. Section 6.2.1 to 6.2.3 focus on the addition of new features/parameters to the sketches on which many different types of CAD features are built. Section 6.2.4 is an enhancement in the definition of multi-section solid feature.

6.2.1 Replacing a 2D straight line with a spline

A straight line is a geometric entity with zero curvature, defined only by its start and end points. The shape of a spline is defined by the position of its control points, which at a minimum must consist of a start and end point, but for which there is no upper limit. Where a spline is only defined by its start and end points (and no additional tangency information is supplied) it will be equivalent to a straight line between those points. For a sketch, the automated approach described herein, will first replace all straight lines in a sketch with a spline, where the start and end points of the straight line are used for the spline curves as shown in Figure 6.1. If the positions of the start and end points are not parameterized, then the approach will also add parameters to their position. It should be noted that replacing a straight line with a spline controlled by two control points does not increase the flexibility of the model, which comes in the next step.

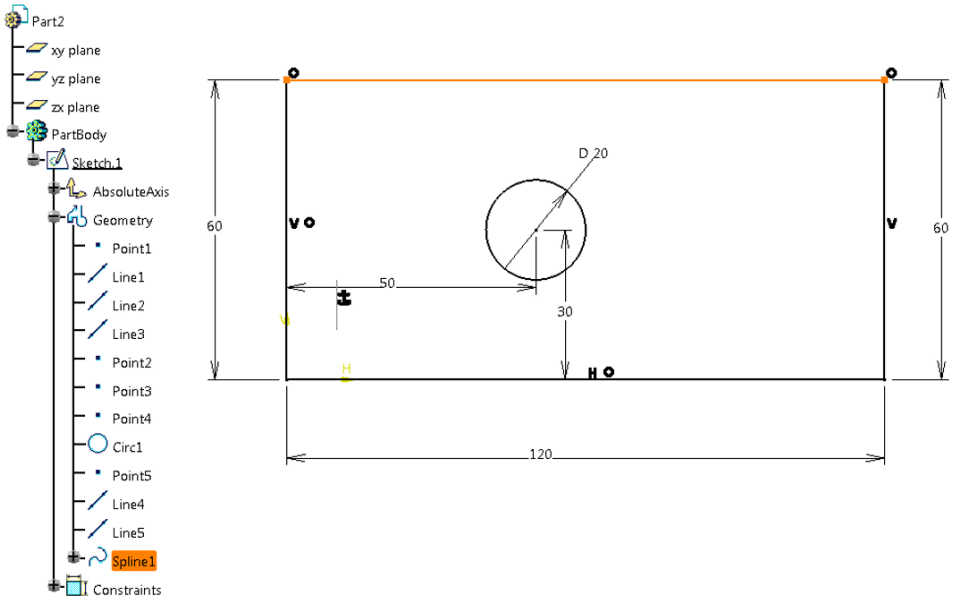


Figure 6.1 2D straight line converted to spline with two control points coincident with end-points of the straight line

6.2.2 Adding additional control points to 2D spline

Increasing the number of control points defining a spline curve increases the design freedom in the curve and allows it to generate more varied shapes. The next step in the process is to automatically insert additional control points into every spline curve in the sketches. New parameters are added to control the values of the *x* and *y* coordinates (on the sketch plane) of the new control points inserted, as shown in Figure 6.2. Once inserted, when the positions of any of the control points are changed, the shape of the spline will update, and any solid modelling feature based upon that sketch will also change in shape accordingly. As such, these new control point parameters have added extra flexibility to the model. The addition of control points to the model can be repeated many times, and in this work, it is limited by the parametric effectiveness of the model.

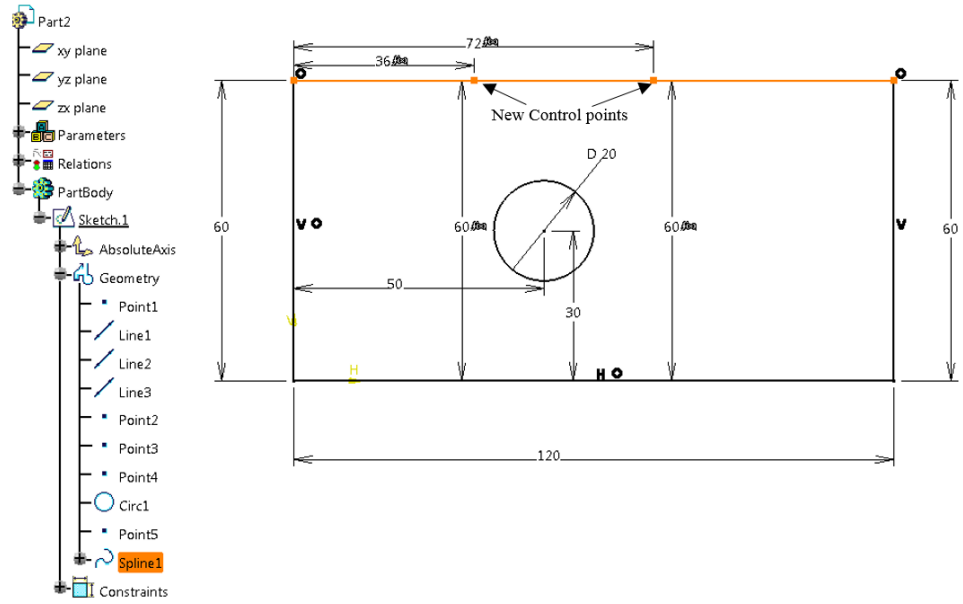


Figure 6.2 Additional control points inserted in 2D spline

6.2.3 Replacing a circle with a spline

A circle is a geometric entity and every point on it is a fixed distance (called radius) from a fixed point (known as the centre point). As such the parameters are the radius and the position of the center point. Enhanced shape variation could be achieved if this circular boundary is changed to a more generalized representation. In this work, this is achieved by re-constructing the circle using a closed B-Spline with 4 control points at angular positions of 90^o to each other (Figure 6.3). The new spline curve is then parameterized using the x and y positions of its control points. It is important to note that this approach should not be employed where it is the design intent to have a circular hole.

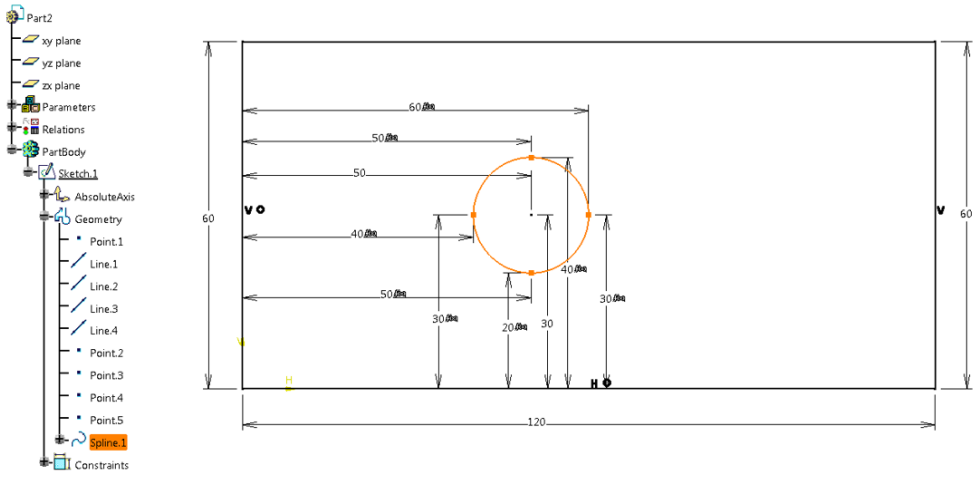


Figure 6.3: Circle converted to spline with four control points

6.2.4 Modifying a Multi-Section Solid

Many components vary in profile along their length (e.g. an aircraft wing or an automotive duct). These components are usually created in a CAD system by creating a number of sketches at different positions and orientations, and then creating a multi-section solid CAD feature (sometimes referred to as a loft) passing through these profiles. These features typically pass through the sketches/profiles exactly, but then interpolate the shape of the feature between the sketches. Guide curves are optional curves that run along the length of the multi-section feature, joining the different profiles, and guiding the interpolation that occurs between them.

One way to modify the shape of the resulting solid features is to use the strategies described in sections 6.2.1 - 6.2.3 to update the shapes of the sketches from which they take their shape. The other approach is to insert additional 2D sketches along the length of the feature, and use the parameters associated with the new sketch as design variables as well. An automated approach was implemented using the following process:

1. The user selects the multi-section feature (containing a region of high adjoint sensitivity) to be updated and defines a point on its surface with the highest adjoint sensitivity.

2. Firstly, the developed approach finds the sub-section of the multi-section solid and creates two planes at the start and end surfaces of the sub-section. A plane can be described as a flat surface that extends without end in all directions. The intersection line between the two planes is computed and a new plane is created using this line and the point defined in step 1. Where the start and end surfaces are defined on two parallel planes, the new plane is created by using the start plane as the reference and passing through the point defined in step 1. The motive to perform this step is to get an appropriate plane, which when intersected with the multi-section feature results in a boundary profile, which is fully constrained within the sub-section being analyzed.

3. A sketch is then created on the plane (created in step 2) using vertex and edges of the boundary profile calculated in step 3 as reference. Here, the straight edges are represented using straight lines with end points on the vertices on the respective edges, and circular arcs are represented using curved edges.

4. If the selected multi-section feature is created using the guide curves, the definition of guide curves is modified to include the sketch vertex in closest proximity ($0.01mm$).
5. Finally, the new sketch profile created in step 4 is included in the definition of multi-section feature.

By following the steps in sections 6.2.1 to 6.2.4 any sketch-based features in the model should now be based on sketches controlled by more parameters. Also, multi-section features should be defined using more sketches than before. It should be noted that at this stage the shape of the CAD model will not have changed, but the additional features and parameters will impart additional freedom to the model.

6.3 Example application: Cantilever Beam

A loaded cantilever beam model is taken as the first test case to demonstrate a methodology for adaptively refining the CAD parameterization during optimization. The design domain is shown in Figure 6.4, and the beam is loaded with a traction force at the right-hand edge. The top and bottom segments of left side are fixed to prevent movement in the horizontal direction.

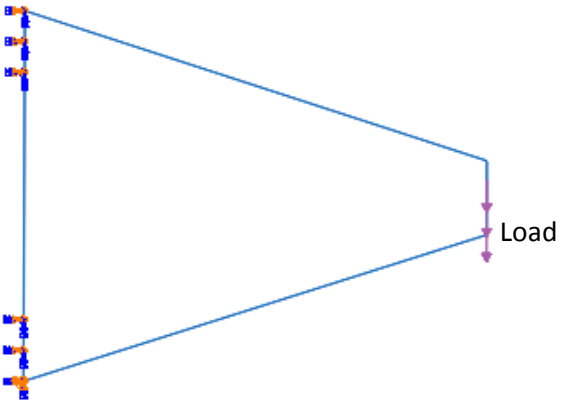


Figure 6.4 Design domain of cantilever beam with boundary conditions

The length of the top and bottom segment of the left edge and the length of right edge are fixed. The bottom left corner of the beam is fixed and restricted to move. Thus, only the sketch lines labelled as A, B and C can be modified during the optimization. The objective function is to minimize the total strain energy while the volume is kept constant. The finite-element mesh generation and analysis of the model is done in ABAQUS CAE, which is linked with CATIA V5 and Scipy optimization modules using python scripts.

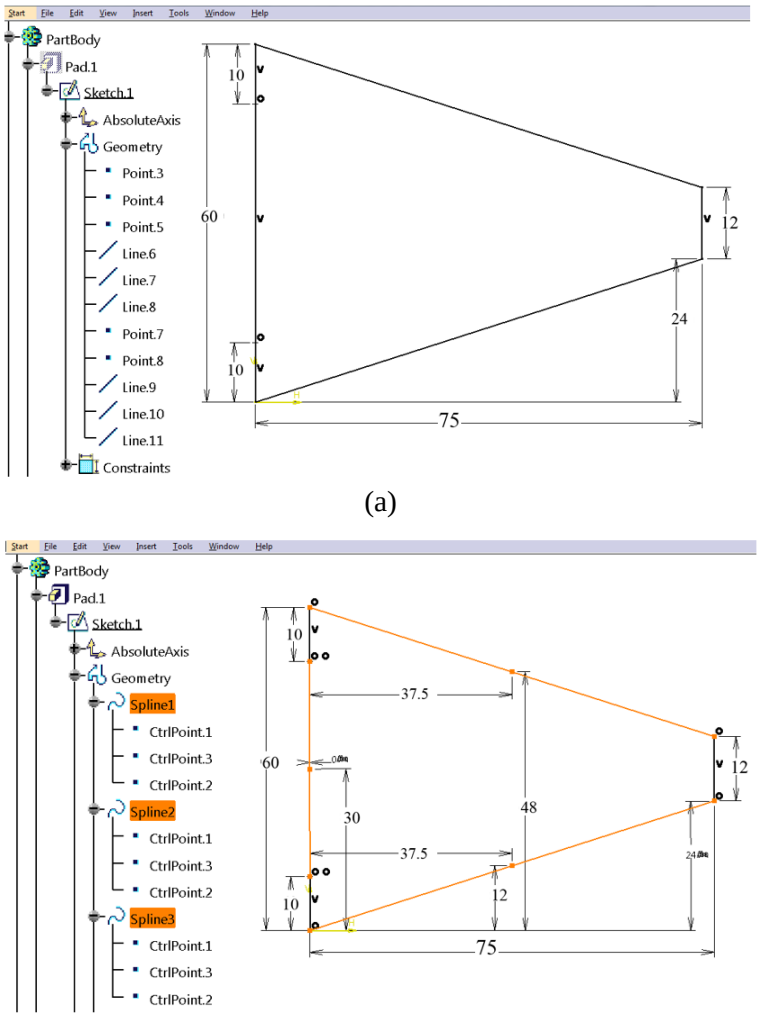


Figure 6.5 CAD 2D sketch (a) with lines, (b) with lines transformed to splines

The design parameters for the initial model are shown in Figure 6.5(a), for which the strain energy contours are shown in Figure 6.6(a), and the total strain energy is calculated as 966 Nm. It is a compliance minimization problem which is a self-adjoint problem having the same primal and adjoint solution. The parametric effectiveness computed for the initial model configuration is 0.0 which indicates that the component cannot be optimized without violating the constraints. Now, the shape optimization was considered employing the automatic feature insertion methodology outlined in section 6.2 to adaptively insert new geometrical entities into the CAD model. When each optimization loop terminates for the current set of parameters, new features and parameters were again added to the model. A constant volume constraint is used to ensure the model did not grow indefinitely. The gradient of this constraint is computed as the change in volume of the beam (computed within CATIA V5) with every parametric update.

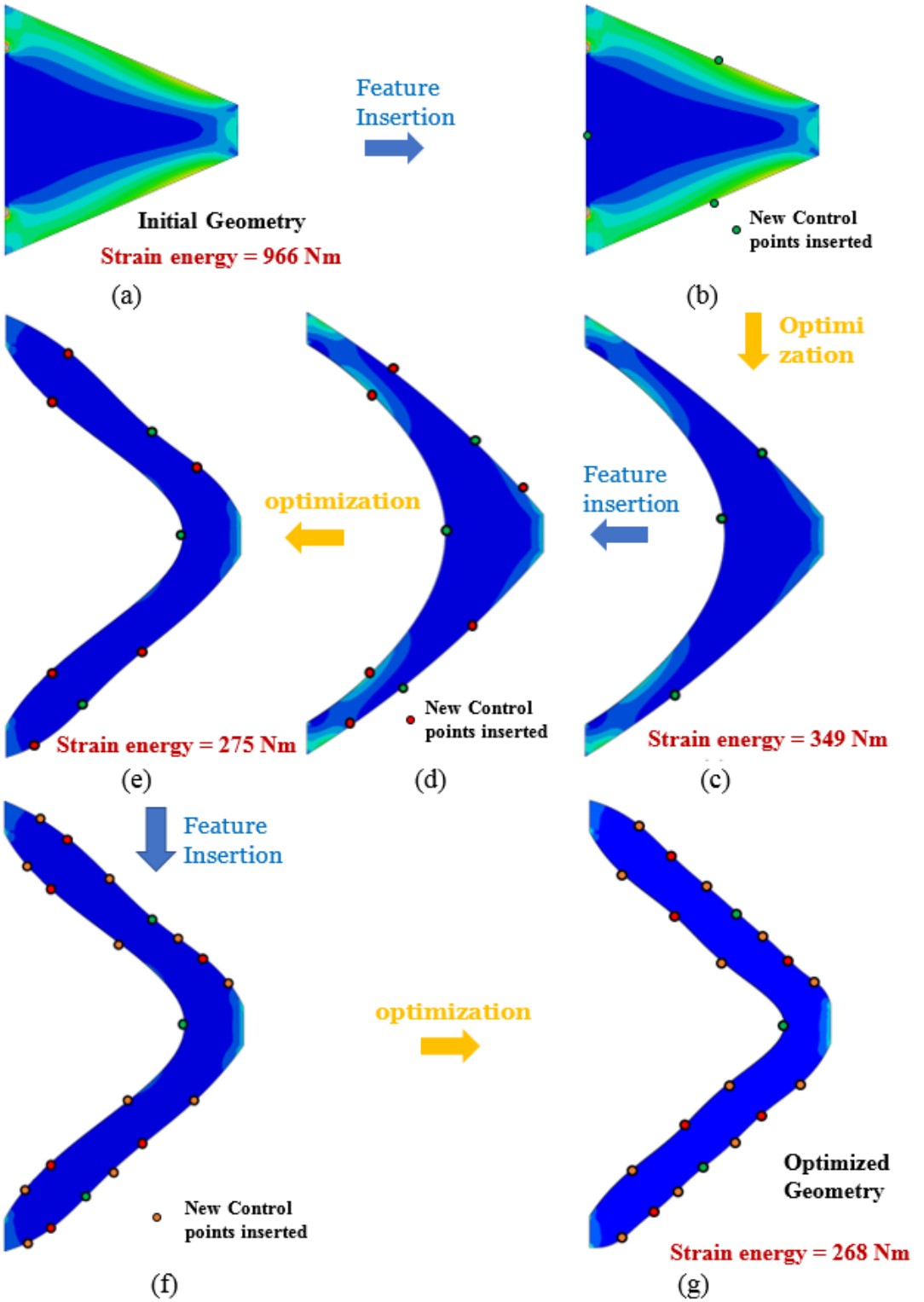


Figure 6.6 Adaptively inserting CAD features during optimization

Firstly the straight lines are transformed to splines each with a control point at its center using the process described in section 6.2.1 and 6.2.2 as shown in Figure 6.5(b), resulting in a parametric effectiveness of 0.24. The position of the new control points is also shown in Figure 6.6(b). The model is then optimized using the SLSQP

algorithm with the constraint of constant volume, resulting in a CAD geometry shown in Figure 6.6(c) with effeciveness reduced to 0.16 and a total strain energy of 349Nm (~ 64 % reduction).

Once the optimizer terminates, additional spline control points are automatically added to the splines (shown in Figure 6.6(d)) to increase the parametric effectiveness to 0.43. A second stage of optimization is run with the new set of parameters resulting in the model shown in Figure 6.6(e) with the total strain energy of 275 Nm (~ 72 % reduction overall) with the parametric effectiveness reduced to 0.10. It can be found that the optimized design is similar to the optimization results obtained in [111, 112]. The overall optimization process with modification of CAD model and the reduction in strain energy is vizualized in Figure 6.6.

6.4 Example application: S-Bend duct

The methodology for inserting new CAD features is applied on the S-Bend duct model analyzed in Chapter 5. Here, a similar modelling strategy is used to create the parametric CAD model in CATIA V5, as shown in Figure 6.7. The duct is modelled using 2D sketch profiles at different positions and orientations, and then developing a multi-section solid passing through these profiles. Similar to the model in Chapter 5, the shape of inlet and outlet duct are fixed, and they are not considered for optimization. This means that for the existing feature sketch 1 and sketch 2, describing the interior profile of the S-Bend can be modified.

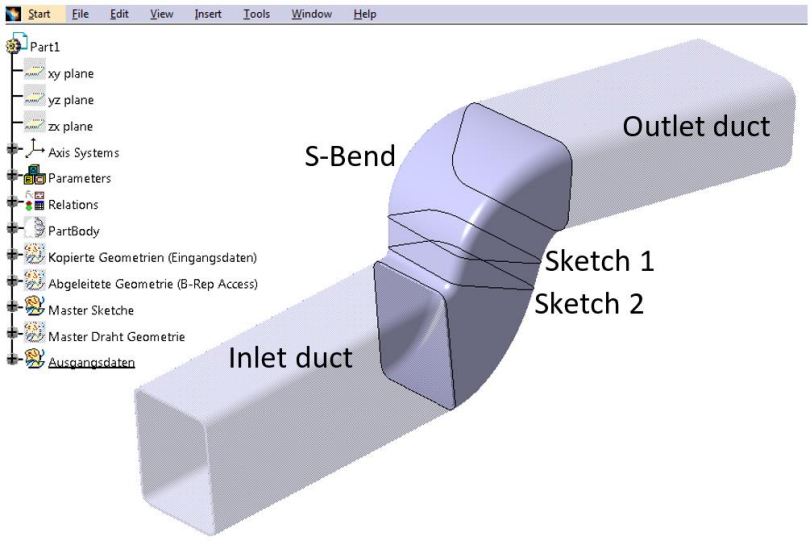


Figure 6.7 CAD model of S-Bend duct created in CATIA V5

The two sketches are parameterized using parameters as shown in Figure 6.8, and orientation of one of the sketch plane is defined by an angular parameter relative to the inlet plane, while the other sketch plane is created at the same orientation offset by a distance. The fluid domain used for CFD analysis is extracted from the body within the CAD environment, by extracting the face patches with tangency constraints. This is done to impose C_1 continuity between the wall faces.

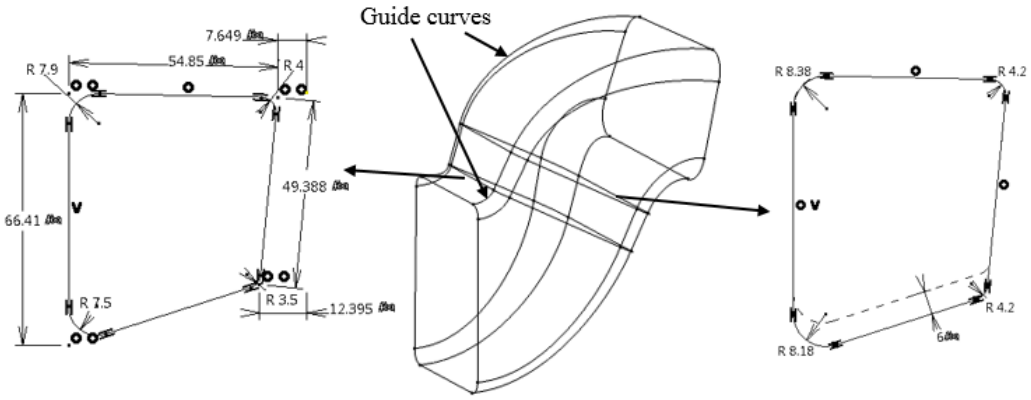


Figure 6.8 S-Bend CAD parameterization

The flow conditions are the same as used in Chapter 5. The objective function considered for optimization is the power dissipation [16] through the duct. The CFD mesh created in ICEM-CFD comprises 486400 elements and is shown in Figure 6.9. The results of primal flow analysis are shown in Figure 6.10, where the pressure and velocity contours are plotted along with streamlines.

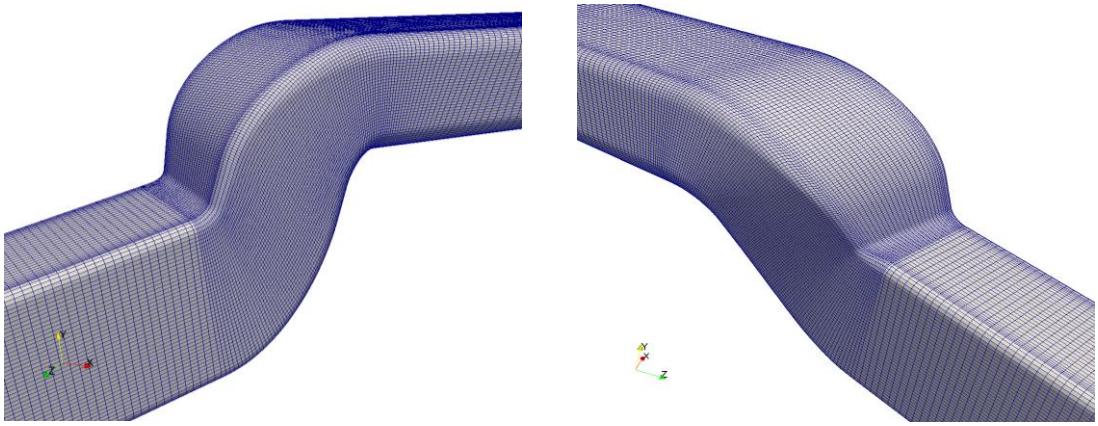


Figure 6.9 CFD mesh for S-Bend

The adjoint sensitivities of the objective function with respect to the normal displacement of the surface are computed at the first optimization cycle and are presented in Figure 6.11. High sensitivities can be seen at the areas of the duct with high curvature. These are the areas in Figure 6.11, where flow separation occurs, contributing to an increase in power dissipation.

The adjoint sensitivities indicate the geometry displacement that would control the flow recirculation and reduce the objective function. It is clear though, the changes in shape suggested by the adjoint sensitivities are rather local and cannot be implemented easily by the original design parameters (see Figure 6.8). The value of parametric effectiveness calculated for initial configuration is 0.14, which indicates the need to add new features to the model. This is achieved following the methodology described in section 6.2.

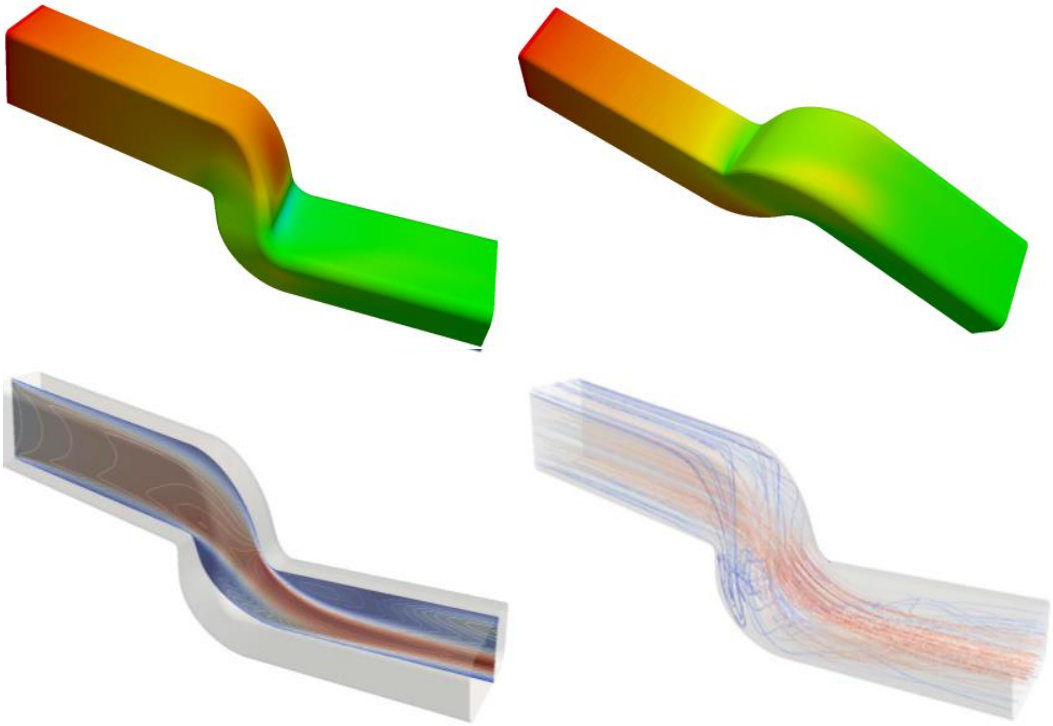


Figure 6.10 Pressure contours (top) and velocity contours and streamlines (bottom) computed for the base geometry

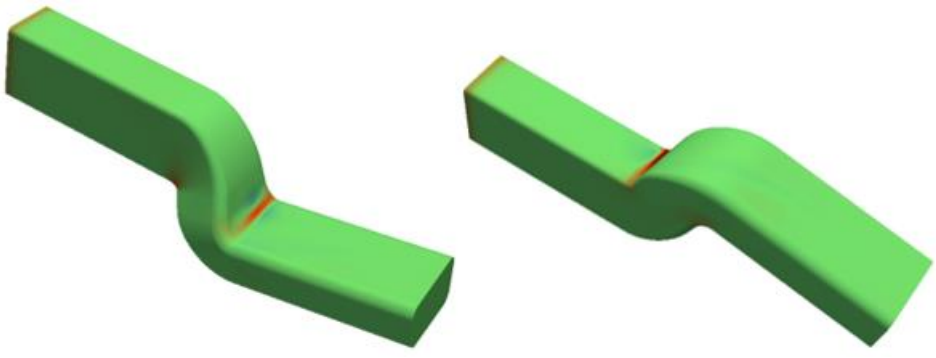


Figure 6.11 Adjoint sensitivities contour. To minimize the objective function (dissipated power) the surface should be pulled out at positive values (warm colours) or pushed in (cold colours)

6.4.1 Replacing 2D straight lines with splines with added flexibility

The first attempt to increase the design flexibility of the S-Bend model was to replace the straight lines in existing sketches with splines and parameterizing their end points. In this case, the radius of circular arcs (defining the rounded corners) and their tangent with the corresponding straight lines are formulated as constraints. A total of eight design parameters are used in addition to the radii of corner circles and tangency between lines and circles to make the sketch fully constrained. An additional control point was added using process described in section 6.2.2, at the centre of each spline in the two sketches as shown in Figure 6.12. This resulted in a total of 32 parameters with a parametric effectiveness of 0.22.

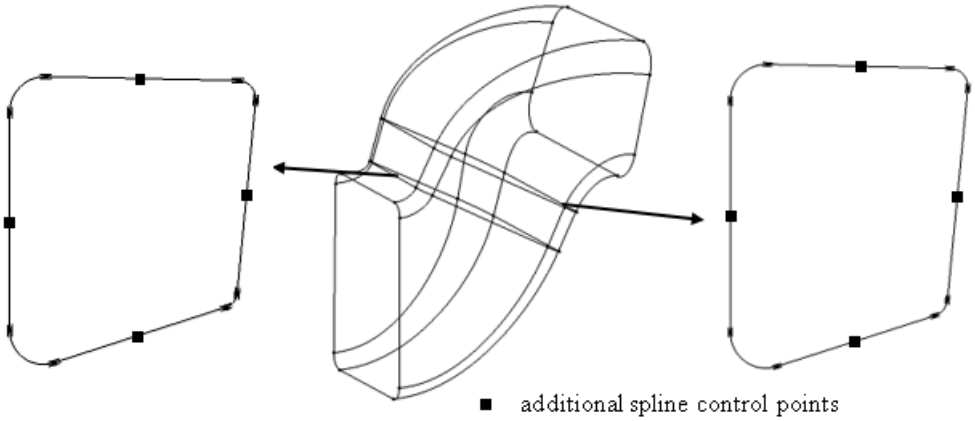
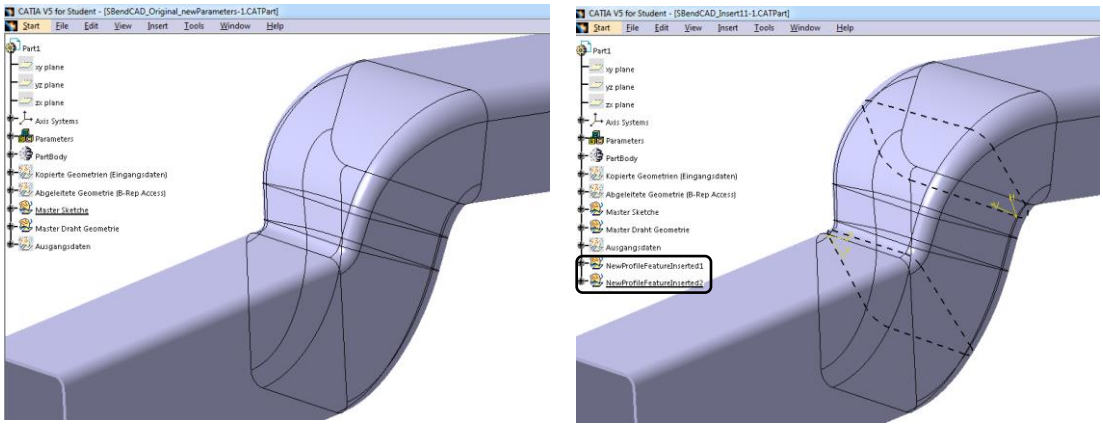


Figure 6.12 Replacing lines with splines with additional control points

6.4.2 Addition of multiple cross-section sketches

As the parametric effectiveness of the model is still low, and since the duct is defined as a multi-section solid, another possible shape modification was to use the process described in section 6.2.4 to insert new 2D sketch profiles along the length of the duct, and use parameters associated with these new sketches as design parameters. It is apparent that the best location to add sketches would be in the areas of high adjoint sensitivity. Analysing the adjoint sensitivity map in Figure 6.11, the two multi-section features defining the two bend areas of the duct were selected and a point was defined on the surface of each feature in the region of high adjoint sensitivity. The new sketch profiles were then automatically inserted and parameterized using the process described in section 6.2.4 and are shown as dashed lines in Figure 6.13. The total number of parameters increased to 48 with parametric effectiveness of 0.61.

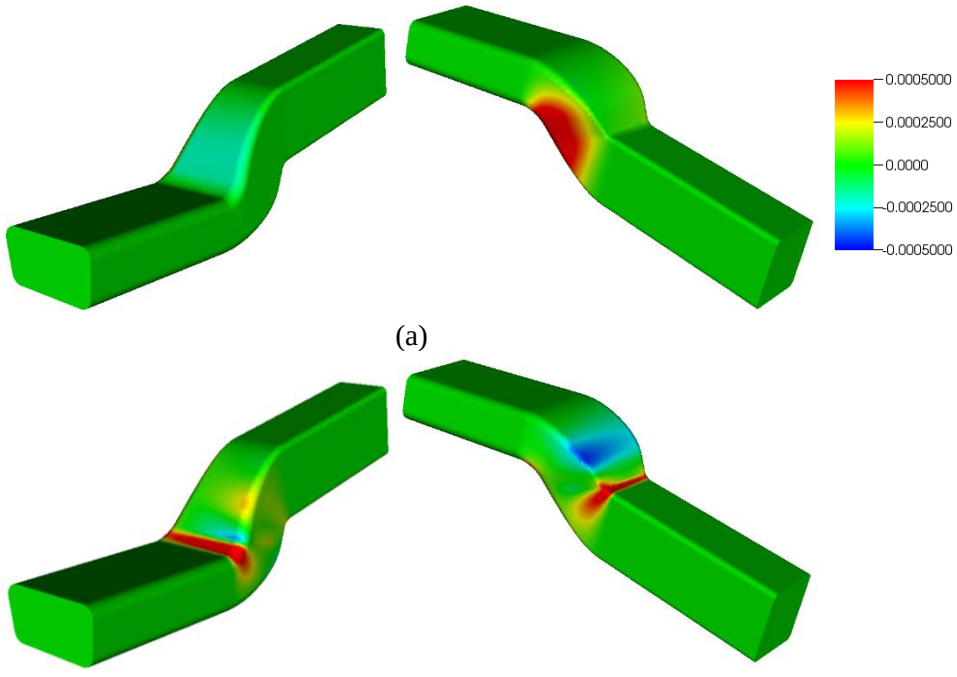


(a) (b)

Figure 6.13 Insertion of new cross-section sketches in the CAD feature tree (a) original CAD, (b) CAD model with new sketches shown with broken lines.

6.4.3 Optimization with new parameters/features

The effect of the increased parametric effectiveness is demonstrated by computing the design velocity contours by perturbing the parameters in the steepest descent direction to achieve the same overall boundary movement. This is done for the initial parameterization created by designer and the parameterization created automatically by inserting features into the CAD model to get the maximum parametric effectiveness.



(a)

(b)

Figure 6.14 Design velocity for the overall boundary movement $dV = 1\ E^{-4}$, (a) initial parameters, (b) with feature insertion

It can be seen in the design velocity contours in Figure 6.14(b) how the new features allow the movement of CAD model boundary in the areas close to the optimum movement suggested by the contours of the adjoint sensitivity map in Figure 5.6, compared to Figure 6.14(a) which shows the much less localised movement given by the original parameters.

The optimization is performed using two different sets of parameterizations i.e. one selected by the designers and the other created by inserting new features into the CAD model feature tree. The insertion of new features was done in three different ways: (1) adaptive parameterization i.e. insert new features when the optimization converges for existing parameterization, (2) progressive parameterization i.e. insert features after fixed number of optimization steps, and (3) static parameterization i.e. insert all new features at the start of optimization. The results of optimization with an objective of minimizing power-loss performed using different methodologies are shown in Figure 6.15. A gradient based optimization method (steepest descent) is employed to minimize the objective function using the CAD parameters as design variables. With each optimization step, the CAD parameter values are updated using a CAD system API developed during this work and for each step a new analysis mesh was created.

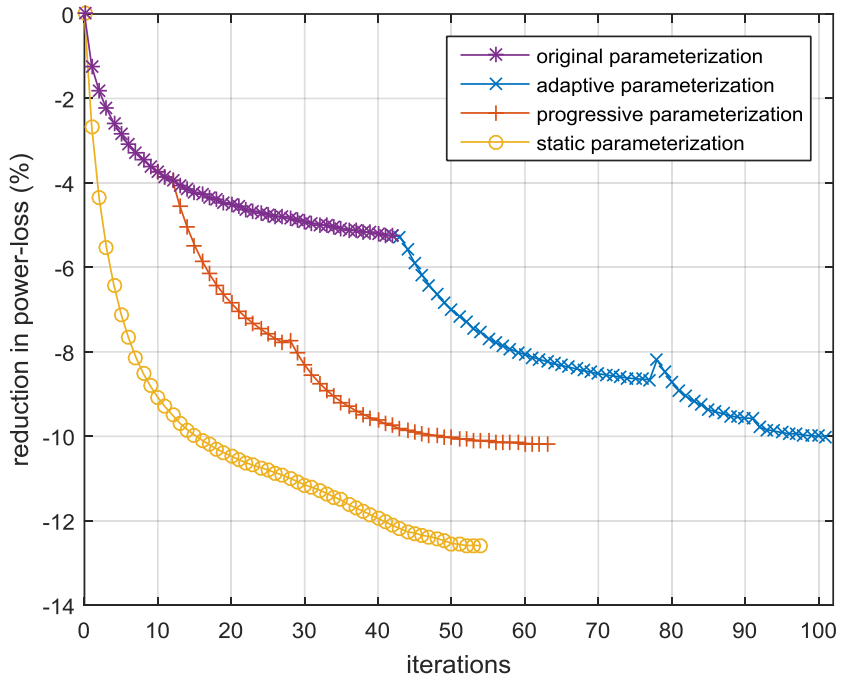


Figure 6.15 Optimization for new CAD features inserted in S-Bend (power-loss as objective function)

The objective function was reduced by 5.25% in 42 steps when optimized using the initial set of design parameters. Following the adaptive and progressive parameterization strategy, the objective function was reduced by 10% in 102 and 63 optimization steps respectively. The insertion of new features at the start of optimization, produced the maximum reduction in objective function by 12.6%. The computational cost for the optimization processes is outlined in Table 6.1.

Table 6.1 Time taken by the optimization process

optimization	Time (approx..)
original parameters	14 hrs
adaptive parameterization	22 hrs
progressive parameterization	29 hrs
static parameterization	19 hrs

The evolution of parametric effectiveness during the optimization process is shown in Figure 6.16, where it can be seen that the parametric effectiveness tends to a small value as the geometry reaches close to the optimum. It should be noted here that the history of parametric effectiveness would allow the termination of the optimization process much earlier than that done by the optimizer which relies on the history of performance reduction. This is especially important when the cost of solution is large.

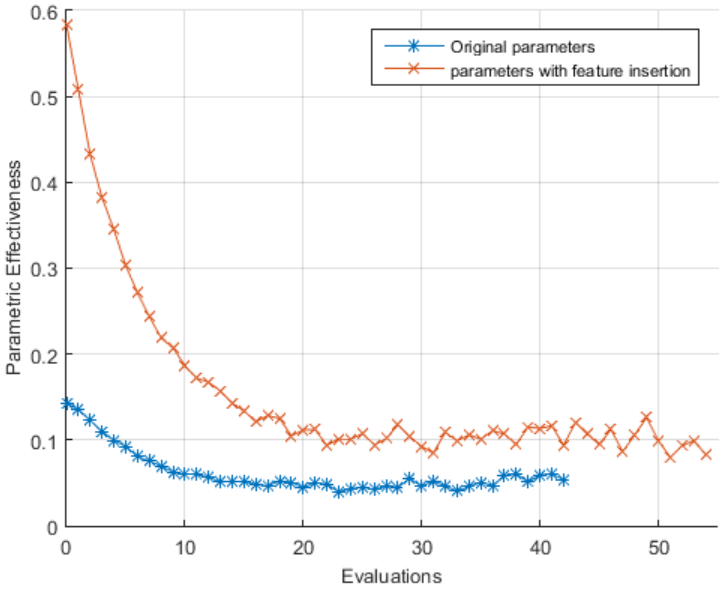
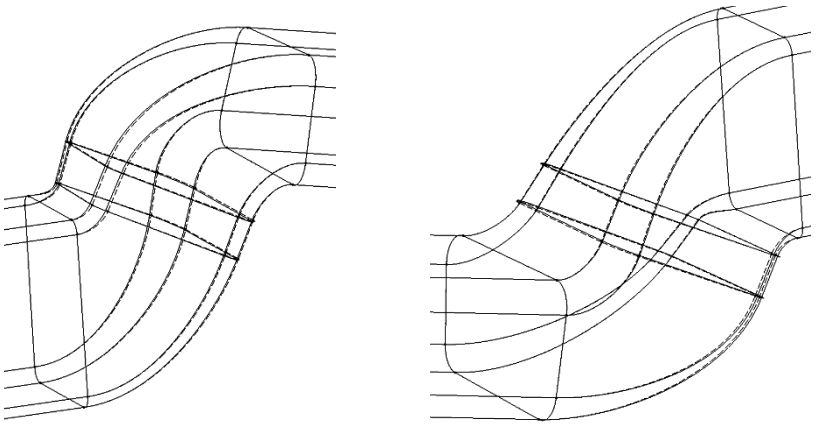


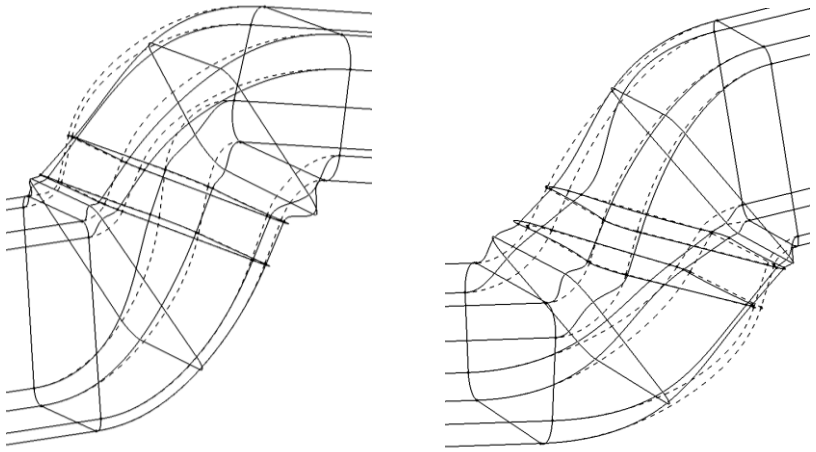
Figure 6.16 Evolution of Parametric effectiveness during the optimization

The comparison between the initial and optimized CAD model geometries for the two parameterization sets is shown in Figure 6.17. The optimized geometry shown in Figure 6.17(b) shows some ripples in the shape near the regions where the S-Bend

section of the duct meets the fixed inlet and outlet duct. On comparing the adjoint sensitivities for the initial (shown in Figure 6.11) and the optimized geometry (shown in Figure 6.18), it can be seen that during the optimization process has significantly reduced the adjoint sensitivities in the bend regions (which were initially highly sensitive), leading to an optimized design which is close to optimum.



(a)



(b)

Figure 6.17 Comparison between original (dashed lines) and optimized (solid line) CAD model for (a) initial parameters, (b) with feature insertion.

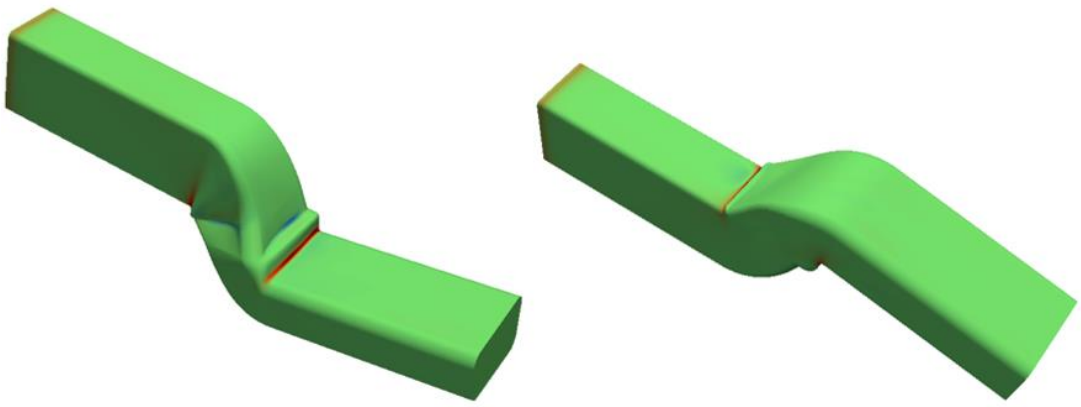
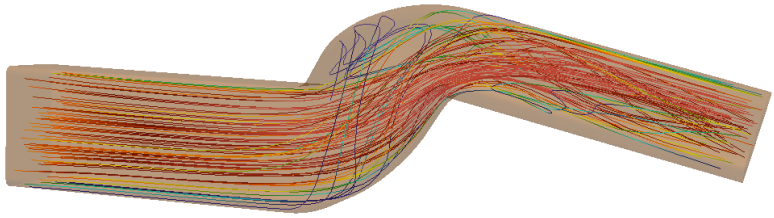
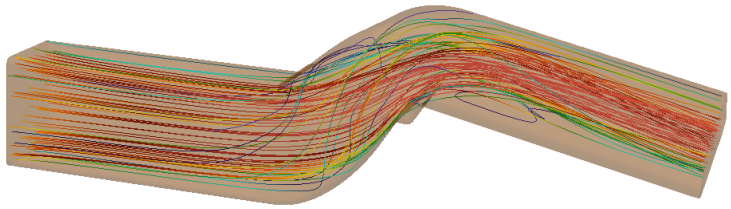


Figure 6.18 Adjoint sensitivities contour on the optimized geometry

The flow streamlines for the initial and optimized geometries are shown in Figure 6.19, where it can be seen that the initial flow field possess greater non-uniformity compared to the optimized flow. The adjoint sensitivities suggested a surface movement that would reduce the high adverse pressure gradients observed on the high-curvature areas of the S-Bend. As the result of optimization, the flow requires less energy to remain attached, subsequently reducing the recirculation region, and thus reduces the objective function. This is seen in Figure 6.20, where the contours of velocity magnitude are plotted at different cross-sections along the duct.

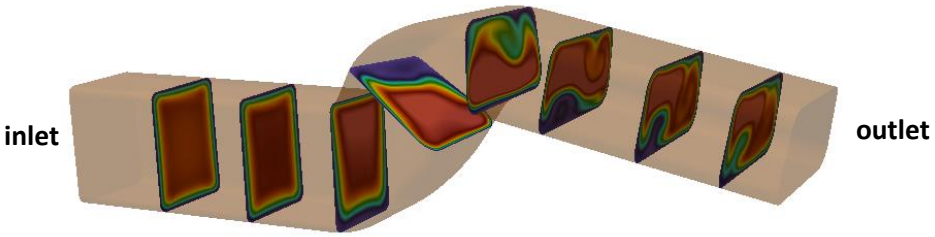


(a)

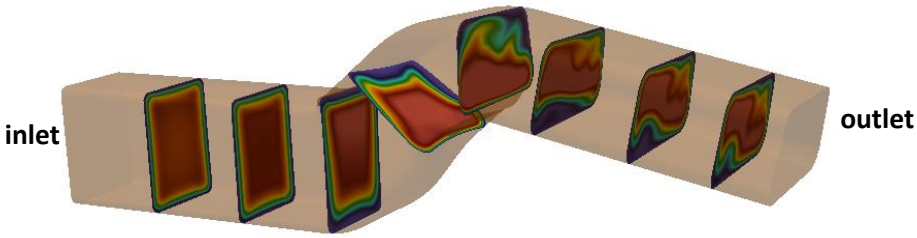


(b)

Figure 6.19 Flow streamlines for (a) initial and (b) optimized geometry



(a)



(b)

Figure 6.20 Contour plots of velocity magnitude for the (a) initial and (b) optimized geometry

6.5 Fitting CAD to mesh

In adjoint based shape optimizations, some approaches use the surface mesh nodes as the design variables, which represents the richest design space available. However, when using a mesh-based method it is the mesh that reaches the optimum and it must be translated into a CAD model before it can be used for further analysis or manufacturing assessments. This mesh-to-CAD step is non-trivial and may require extensive user interaction [52, 53]. In this section, an approach is described which attempts to fit a CAD model to an optimized mesh. It is setup as an optimization problem where the parameters defining a CAD model are used as design variables and the objective function is to minimize the distance between the CAD boundary and the mesh obtained using node-based adjoint optimization technique. This approach is similar to the inverse design methods used for aerodynamic design optimization where the objective function was defined to minimize the difference between the initial and optimum pressure distribution at the surface mesh points [113-116]. The benefit of this approach is that it allows exploration of a large design space (mesh nodes as design variables) for flow optimization, and subsequently fitting a feature-based CAD model to the optimized mesh resulting in a CAD geometry at the end of the process which can be used for downstream applications.

The objective function is mathematically formulated as

$$J = \sum \left[X_{target} - X_{CAD}\right]^2, \tag{6.1}$$

where, X_{cad} and X_{target} are the points on the CAD model and the optimized mesh respectively. The term in the right-hand side of Eqn. 6.1 is simply the square of design velocity (Chapter 3) computed between a mesh and the facet representation of the CAD model. The gradient of objective function with respect to CAD parameters (θ) can then be calculated as:

$$\frac{dJ}{d\theta} = -2 * \sum \left[X_{target} - X_{CAD}\right] * \frac{dX_{cad}}{d\theta} \tag{6.2}$$

The objective function and the gradients can now be used within the gradient based optimization framework (Chapter 4) to get an optimized CAD model which is as close to the shape of the optimum mesh as can be achieved using the parameters. The success of this approach is highly dependent on the CAD features used to construct the CAD model. Herein, the methodologies developed earlier in the chapter are used to increase

the design freedom of the model by inserting new features to the CAD model feature tree. On the contrary, to the adjoint sensitivities driving the feature insertion process, here the design velocity contours are used to decide where the new features are to be inserted.

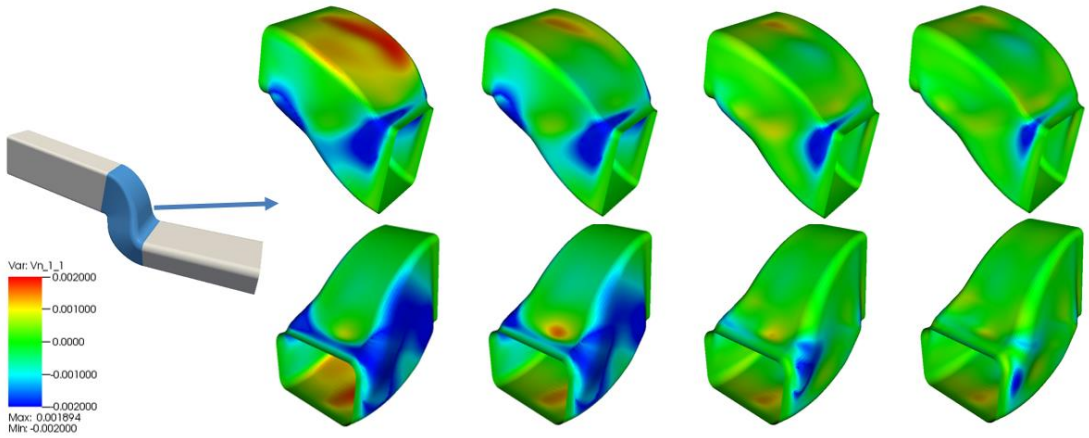
6.5.1 Example Application

A CAD-free adjoint optimization (using volumetric B-Splines [117]) is performed to minimize the power loss across the S-Bend duct shown in Figure 6.7. Here, the mesh from the optimization process is selected and stored in STL (STereoLithography) format. Now the design velocity ($X_{target} - X_{CAD}$) is calculated using the approach outlined in Chapter 3 and used to compute the values of objective function and gradients from Eqn. 6.1 and 6.2 respectively.

The difference between the CAD model and objective mesh is shown as a measure of design velocity for the various stages of optimization of the CAD model in Figure 6.21. As shown in the design velocity contours for the initial CAD geometry, in Figure 6.21(a), there is a significant difference between the CAD and the optimized mesh. Firstly, the CAD model was optimized using the initial set of parameters described in Figure 6.8, to achieve an optimized geometry with a reduction of 2.35% in the power-loss defined as the objective function. The design velocities comparing the optimized CAD with the target mesh is shown Figure 6.21(b). Now, additional cross-sectional sketches are inserted using the methodology described in section 6.5.2 and subsequently optimized with the new set of CAD parameters to obtain a reduction in objective function of 6.1%. The decrease in objective function is complemented by the observed design velocity contours in Figure 6.21(c), where significant change is observed compared to the initial models. Since the design velocity contours showed that more flexibility is needed in the areas where the new sketches were inserted in the last optimization cycle, so in the next step, the methodology in section 6.5.1 is used to convert the straight lines in the inserted sketches to splines with an additional control point to further optimize the geometry. The design velocity contours on the final geometry are shown in Figure 6.21(d).

The geometry optimized using the inverse design method showed a reduction in objective function by 7.2% compared to 8.29% achieved with mesh-based optimization method. Although the approach could not completely match the CAD

model to the given mesh, it did provide a CAD model which has a much better performance.



(a) (b) (c) (d)

Figure 6.21 Fitting CAD to mesh for S-Bend

6.6 Summary

The objective of this chapter was to describe an approach to increase the design flexibility of a feature-based CAD model. The approach of parametric effectiveness as described in Chapter 5, was used as an indicator to decide when to insert new CAD features. It was argued that the design decision on which CAD feature to insert, and where, can be made by analyzing the adjoint sensitivities. The insertion of new CAD features was performed in an automated way using Python-CAD system APIs developed during this research. The effectiveness of inserting CAD features was calculated using a single adjoint analysis, thus utilizing the same computational resource to analyze multiple features. A comparison of parametric effectiveness and the optimization results for a number of CAD models was presented where different feature update strategies were applied.

The results were presented for the optimization of a cantilever beam model, where the objective function was to minimize the total strain energy. For another test case, the shape of an S-Bend duct was optimized to reduce the power-dissipation across the duct. In both the cases, the optimization performed by inserting new CAD features showed significantly higher performance gains. The application of the feature insertion methodology was also demonstrated for fitting a parametric CAD model of an S-Bend duct to an optimized mesh obtained using CAD-free optimization methods.

Chapter 7 CAD-based adjoint optimization with assembly constraints

7.1 Introduction

In general, mechanical design processes are not only driven by performance but are subjected to constraints. Some of these constraints include: the trailing edge radius of turbine blade, volume constraint, constraint on cross-sectional area, constraint on flow fields to account for a minimum lift, fixed exit-flow angle etc. Mader and Martins [118] used constraints such as bending moment, static and dynamic stability to examine optimal wing shapes in subsonic and transonic flows. Walther and Siva [119] presented an adjoint-based shape optimization for a multistage turbine design, with the objective to maximize the efficiency while constraining the mass flow rate and the total pressure ratio. Kontoleontos et al. [120] presented a constrained topology optimization approach for ducts with multiple outlets. Here, the flow constraints are in-forced at each outlet defining the volume flow rates, flow direction and/or mean temperature of the outgoing flow. In terms of geometrical constraints, Shenren et al. [59] presented an approach employing a set of test points to impose the thickness and trailing edge radius constraint for the optimization of a nozzle guide vane. The presence of constraints often restricts the shape optimization methods to be applied to a few selected applications, where constraints do not dominate or can be accounted for post-optimization.

When optimizing an industrial design, one of the important factors to be considered is the packaging space in which the optimized component is expected to fit, which is typically defined by other components in the assembly and defines the regions the component being optimized is not allowed to violate. Since different components are designed and optimized by different designers, when the components are assembled together, issues such as fit often occur, requiring engineering changes late in the product development cycle [17]. Thus, it is important for designers and manufacturers

to devise methods to ensure that the optimized component can be assembled within the space available before the actual component is manufactured. With the advances in CAD systems and development of Digital Mock-Ups (DMUs) for complex CAD model assemblies, it is now possible to replace the physical prototypes with virtual ones and do the assembly of components in a virtual environment before any prototype is built. The DMU is a product assembly workbench where different components are positioned in 3D space relative to each other. Interference can occur during assembly when two or more components are designed such that they attempt to occupy the same physical space when assembled. An obvious solution to address the aforementioned limitations would be to include constraints imposed by adjacent components of the assembly during the design optimization of individual components or apply class detection and fix the interferences during the product assembly.

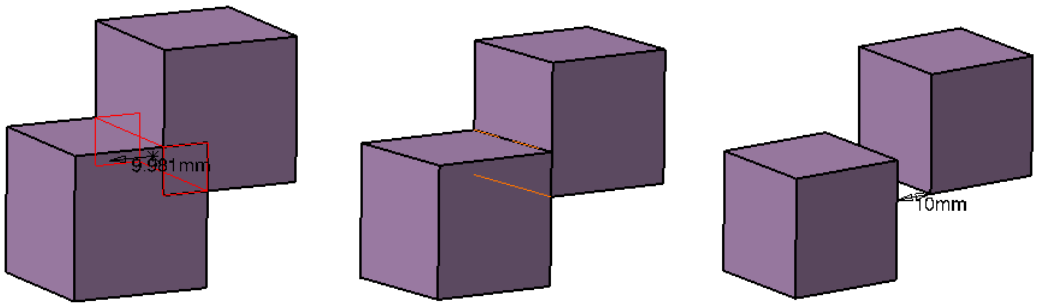
Some of the early works in the field of interference detection between two solids were found in [121, 122]. Recent developments in this field include [123], which enabled interference detection directly using CAD models. Zubairi et al. [124] developed a sensitivity approach to eliminate interference (if present) in a 3D CAD assembly, by identifying which parameters defining the CAD features needs to be modified and by how much to eliminate interference. The approach is effective in this role (eliminating interferences), but the effect of the resulting shape change on the performance of the individual components was not considered, meaning that the process of eliminating interference could also reduce the performance of a products, or even make it unsuitable for its role. Recently, a 2D shape optimization of NACA0012 based on polynomial response surface model was presented in [125], where assembly constraints were formulated to consider the presence of a fuel box in the interior of the wing profile. The constraints were enforced as constant upper and lower bounds for the design parameters. Although the approach was successfully applied for optimization, using only the parameter bounds may be highly restrictive when the number of design variables is increased.

This chapter uses the approaches developed in Chapter 3 and Chapter 4 along with further developments to optimize a component in terms of its performance, but with the assembly constraints imposed on the system. The developed approach is demonstrated on 2D and 3D parametric CAD models built in CATIA V5, assembled with other components defined in CATIA V5 assembly workbench.

7.2 Interference detection

Interference occurs when some components in an assembly violate others by occupying the same physical space. The interference detection system in CATIA V5 provides capabilities to obtain the penetration depth between the interfering components, which is described as the minimum distance required to translate a product to avoid interference. In addition, the clearance distance between two components can also be obtained.

Figure 7.1 displays the part-to-part interference detection tool in CATIA V5, which shows if the selected parts are interfering or are in contact or have a clearance between them. A CAD system API is developed in this work to automatically calculate the interference between the component being analyzed and other assembly components. In this chapter, a positive value of interference defines the minimum distance of clearance between the parts, while a negative value defines the amount of maximum penetration distance between the parts.



(a) (b) (c)

Figure 7.1 Interference between two boxes as in CATIA V5, (a) Interference, (b) Contact, and (c) Clearance

The first requirement from the perspective of optimization is to know the number of components in the CAD assembly and compute the amount of interferences between them. This is achieved by using a CAD system API which is configured to detect the components (other than the component being optimized) in the CATIA product assembly module (i) and use the interference tool to compute the individual interferences with the initial CAD model ($\delta_i^{initial}$). At each optimization step, the developed CAD system API records the name of different components in the CAD assembly and the obtained interference values (used as assembly constraints).

The other requirement is the computation of gradients of each assembly constraint with respect to the parameters used to define the initial CAD model. CATIA V5 offers capabilities to access the part model's parameterization through the assembly workbench. So, to compute the gradients of constraints, each parameter of the CAD model is perturbed by a small amount ($\Delta\theta_i$), and the interference tool is used to obtain the new interference values ($\delta_i^{updated}$) for all components. The respective gradients of the constraints are then obtained using a finite difference method as

$$gradient = \frac{\delta_i^{updated} - \delta_i^{initial}}{\Delta\theta_i} \tag{7.1}$$

Here, the approach for selection of step-size to perturb CAD parameter is the same as that used for computing the design velocities in Chapter 3.

7.3 Optimization framework

A gradient based optimization technique is used to guide the design towards a local optimum over multiple optimization steps. A general optimization with assembly constraints can be defined as:

$$\begin{aligned} &\textit{Minimize:} && \textit{objective function,} \\ &\textit{Subject to:} && \textit{interference} < 0 \\ &\textit{design variables:} && \textit{vector of CAD parameters} \end{aligned} \tag{7.2}$$

For the process of optimization, the CAD centric optimization approach presented in Chapter 4 is followed. The constraints due to interference with the adjacent components in the product assembly are used to enforce an inequality constraint for the optimizer such that the value of interference is less than zero. The flow diagram for the optimization process is shown in Figure 7.2.

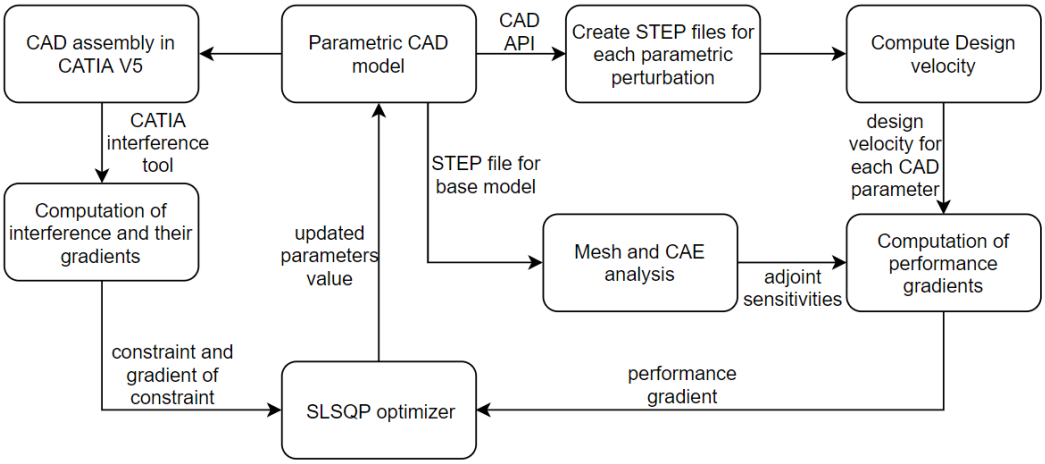


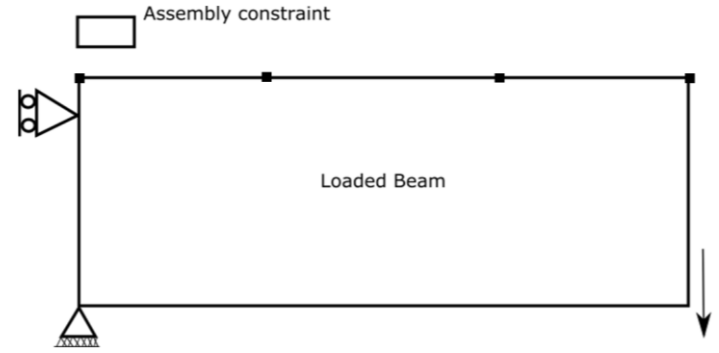
Figure 7.2 CAD-based optimization using constraints from assembly components

The optimization framework is configured such that it runs the CAD system API to automatically update the parameter values and subsequently use the updated part model in the assembly workbench to compute the interference from other components. A similar CAD system API is configured to perturb the parameters of the part model and compute the change in interference to yield the gradients of constraints to be used for optimization.

7.4 Example applications

7.4.1 Cantilever Beam Optimization

A simple test case of a cantilever beam loaded at one end is considered to demonstrate the applicability of the optimization framework incorporating the assembly constraints. The beam's geometrical configuration, the loading applied, and boundary conditions are shown in Figure 7.3 (a). The top edge of the beam is defined by a Bézier curve with four control points, while the bottom, left and right edges are defined using straight lines. The beam is modelled in CATIA V5. In the initial geometrical configuration, the strain energy density (adjoint sensitivity) is higher at the left-hand corners of the beam as shown in Figure 7.3 (b). This means that when minimizing strain energy, the geometry is expected to move outward in that region. A constant volume constraint was also imposed for the test case to ensure the model did not grow indefinitely (as an objective of minimizing compliance would encourage).

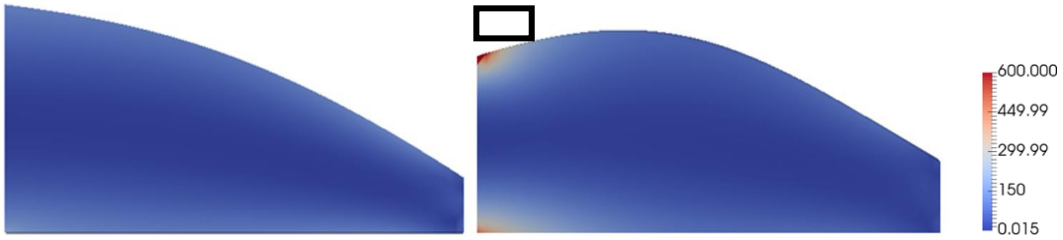


(a)

(b)

Figure 7.3 (a) Cantilever beam with boundary conditions, (b) strain energy density

The optimization of this component was carried out twice. For the first optimization there was no constraint imposed on the packaging space for the component. For the second optimization a rectangular box was added representing an adjacent component, restricting the amount of outward movement possible by the top edge, Figure 7.3 (a).



(a) (b)

Figure 7.4 Optimized cantilever beam with (a) constant volume constraint, and (b) assembly constraints

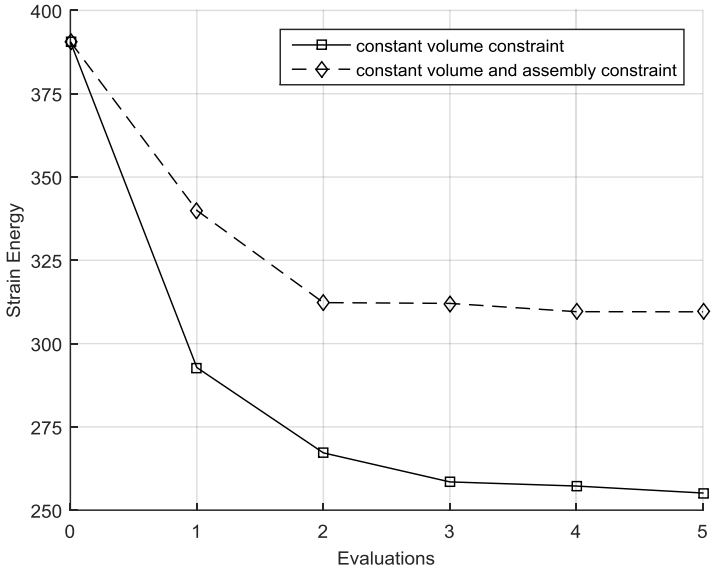


Figure 7.5 Optimization history for cantilever beam

The optimization results are shown in Figure 7.4 and Figure 7.5. The optimization without the constraint on packaging space, Figure 7.4(a) has resulted in the expected thickening of the left-hand side of the beam, and a subsequent narrowing of the right-hand side to maintain the overall volume of the model. It is obvious that this has caused the boundary to move outwards in the regions of highest strain energy density (remembering that the bottom edge is constrained to be a straight line). In the other optimization, Figure 7.4(b), it is apparent that the outward movement of the model is restricted due to the presence of the block component. As a result, the optimizer finds a different solution and as shown in Figure 7.5, this results in comparatively lower reduction in the strain energy of the beam. It should be noted that the optimized model

in Figure 7.4(a) would have interfered with the block component by approximately 8mm.

7.4.2 NACA 0012 optimization

The applicability of the developed framework has also been demonstrated the 2D NACA0012 aerofoil, which has also been analyzed in Chapter 4. The assembly constraints considered here are representative of the presence of a rectangular fuel-box, to be contained within the aerofoil, as represented by the broken line in Figure 7.6. Here, the 2D model was extruded by ± 1mm to create 3D geometry which is then used to compute clash in the product assembly workbench.

The flow conditions are defined as:

- Freestream Temperature = 273.15 K
- Freestream Mach number = 0.85
- Angle of attack (α) = 0^o
- Objective Function = $min(C_d)$
- Design variables = 5

The CFD mesh is created in ICEM-CFD using a multi-block strategy with 300 points along the aerofoil and 51 points in the direction normal to the aerofoil. A detailed view of the mesh around the aerofoil is shown in Figure 7.7(a). A strong shock-wave is formed at upper surface of the aerofoil as shown in Figure 7.7(b) which contributes to increased drag on the aerofoil.

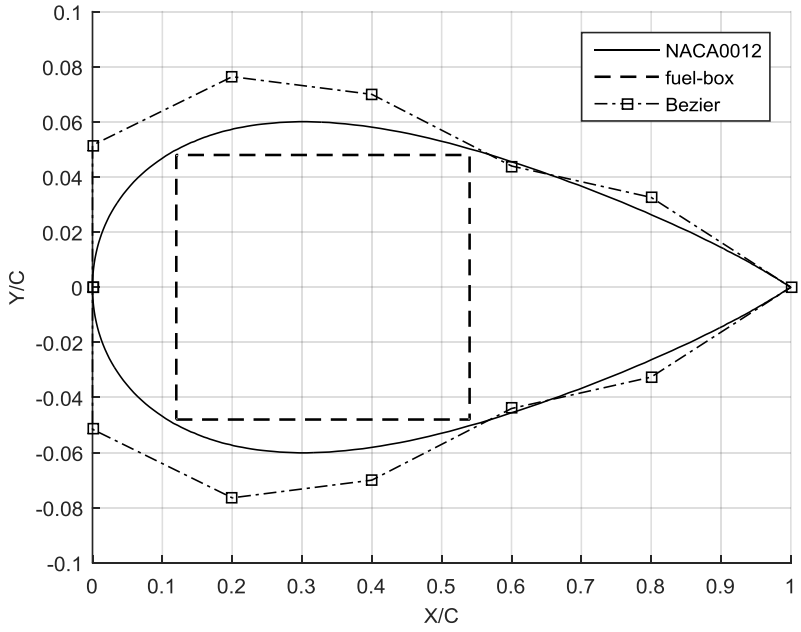
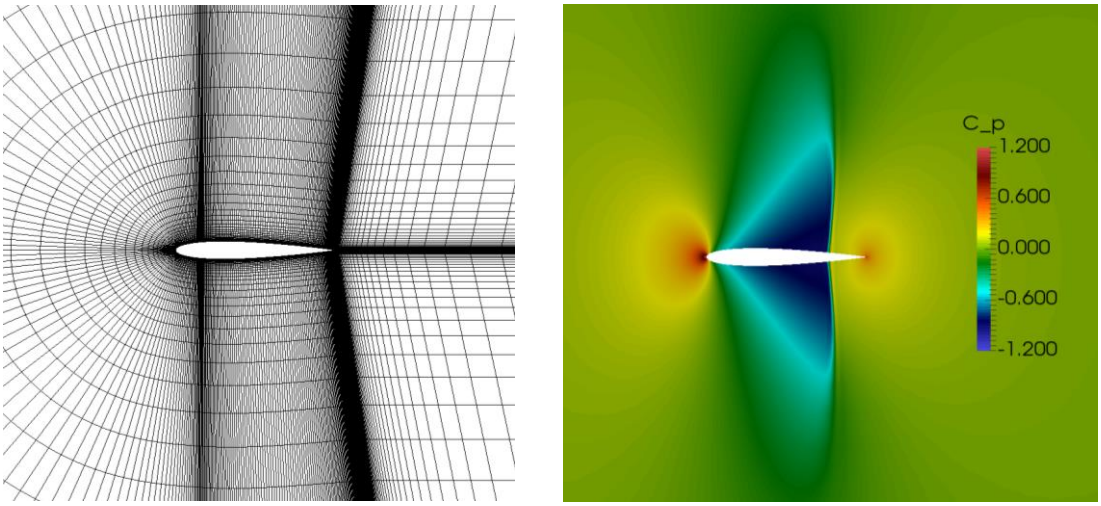


Figure 7.6 NACA0012 aerofoil with Bezier control points and fuel-box



(a) (b)

Figure 7.7 (a) Mesh around NACA0012 aerofoil. (b) C_p distribution at the start of optimization

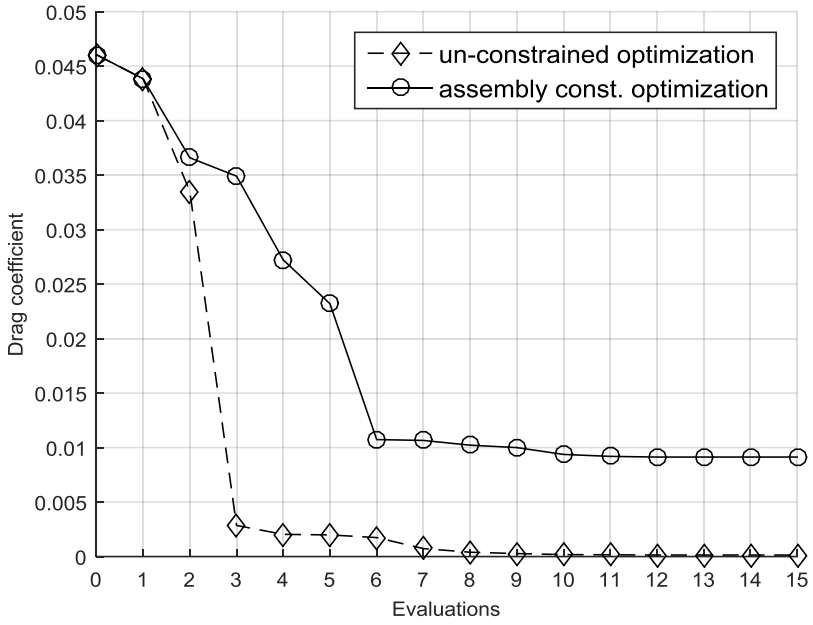


Figure 7.8 Optimization history for NACA0012 with assembly constraint

As with the cantilever beam, this optimization was carried out with and without the constraints imposed by adjacent components. For each optimization step, a new CFD mesh is created in ICEM-CFD using an automated blocking script. The optimization history of the aerofoil with assembly constraint is shown in Figure 7.8, where it is compared against the optimization when no such constraint exists. The drag coefficient is reduced from 0.04605 to 0.0091 with the assembly constraint enforced compared to 0.00013 for an un-constrained optimization with parameter bounds. The optimized geometries are compared in Figure 7.9, where the Y-axis is amplified to enhance visual comparison. The un-constrained optimization results in a thinner aerofoil, compared

to that obtained in the presence of assembly constraints. The pressure distributions around the aerofoil are shown in Figure 7.10.

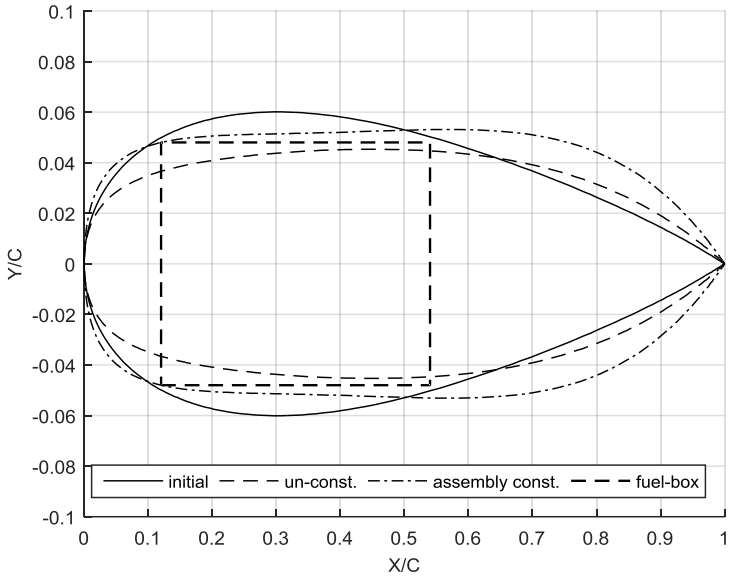
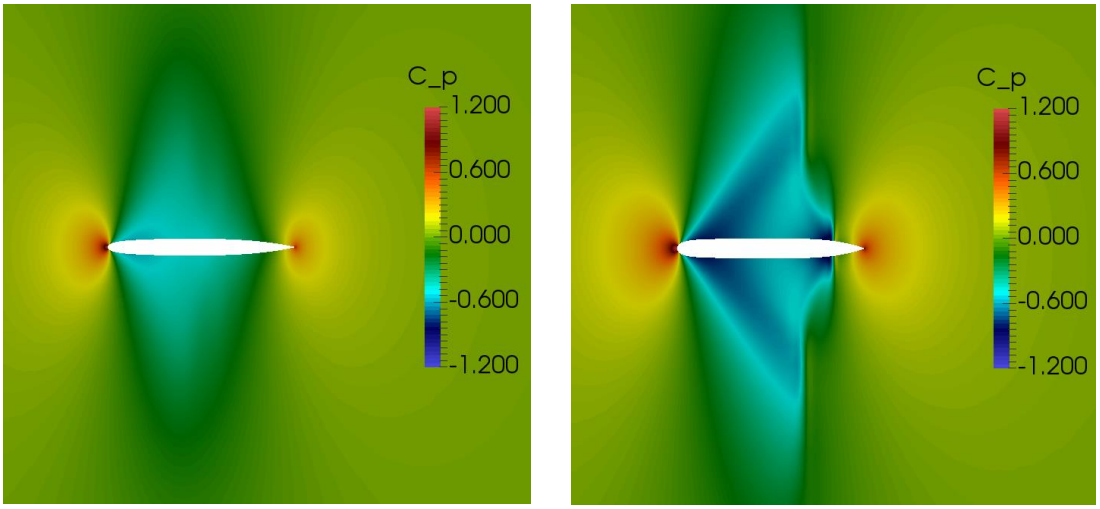


Figure 7.9 NACA0012 aerofoil optimized with assembly constraint



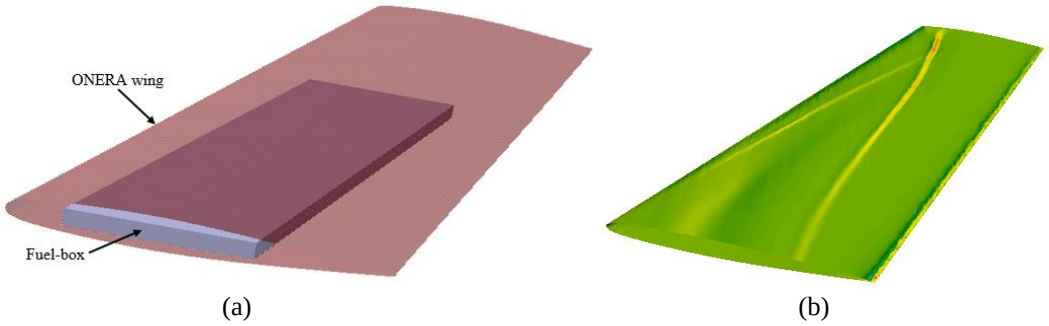
(a) (b)

Figure 7.10 pressure contours on optimized NACA0012. (a) un-constrained, (b) with assembly constraint

7.4.3 ONERA M6 wing optimization

The other test case is the 3D ONERA M6 wing model, which has been described in Chapter 4. Here, a 3D rectangular fuel-box needs to be contained within the wing volume as shown in Figure 7.11(a). For flow analysis, an unstructured mesh was created in GMSH and consisted of 150,410 nodes and 767,450 tetrahedral elements. The adjoint sensitivity for minimizing aerodynamic drag over the ONERA M6 wing

is shown in Figure 7.11(b), which indicates how the mesh nodes on the wing's upper surface should move to minimize the drag.



(a) (b)

Figure 7.11 ONERA M6 wing (a) with fuel-box, and (b) adjoint sensitivity plot

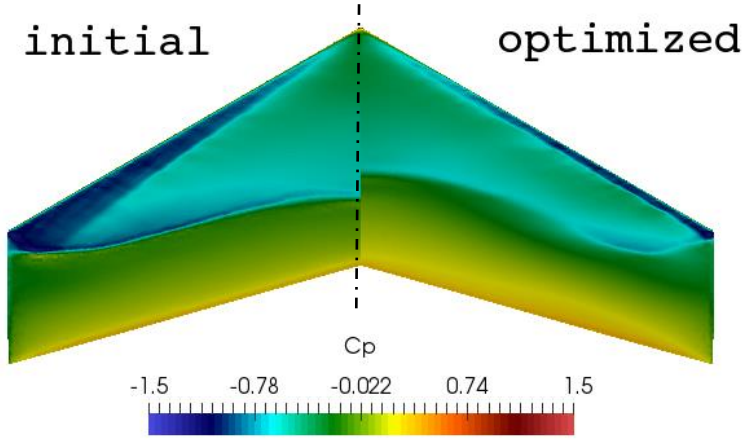


Figure 7.12 C_p distribution on ONERA M6 wing, initial (left) and optimized (right).

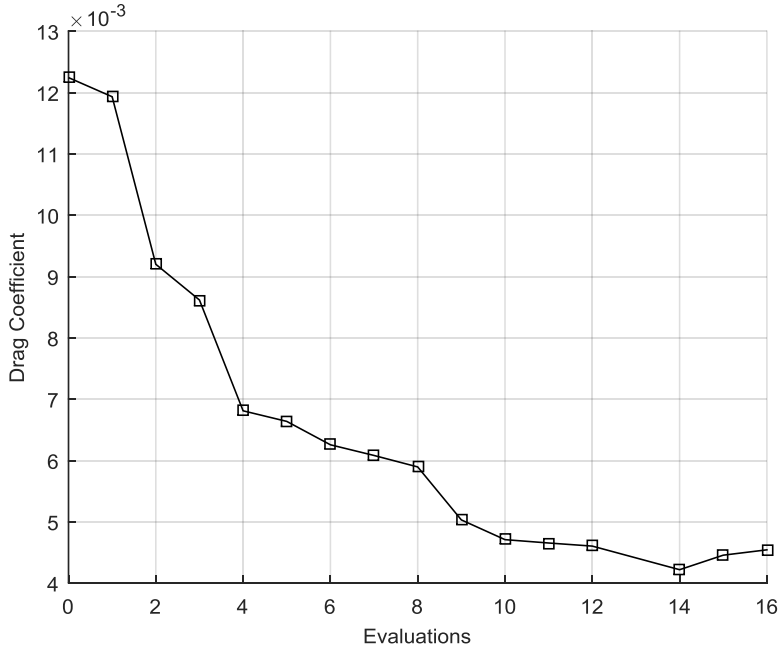


Figure 7.13 Optimization history for ONERA M6 with assembly constraint

For each optimization step, a new mesh was created in GMSH using the density box and background field functionality in GMSH. The pressure distribution on the initial

and optimized ONERA M6 wing model is shown in Figure 7.12. During the optimization the drag coefficient was reduced from 0.01232 to 0.0045 as shown in the optimization history plot in Figure 7.13. A comparison between the initial and optimized geometry at two different cross-sections is shown in Figure 7.14 and Figure 7.15 respectively, where the vertical axis is amplified to enhance visual comparison. In each case, the optimized result is very close to the contact with the fuel box constraint.

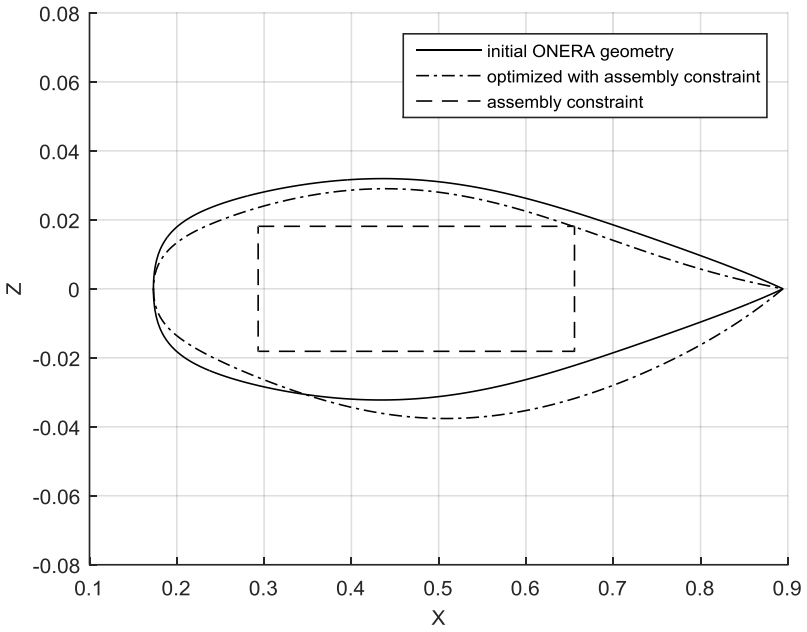


Figure 7.14 ONERA M6 wing optimized section at 30% span

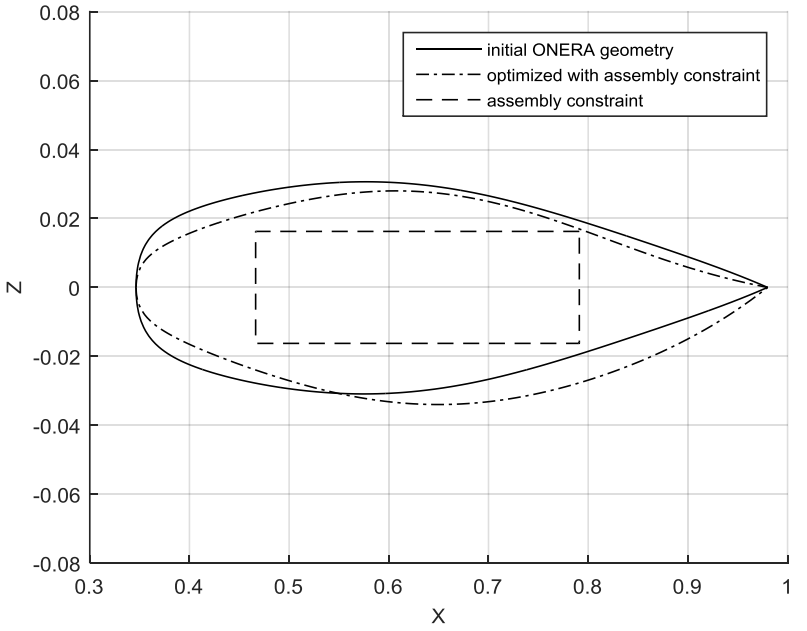
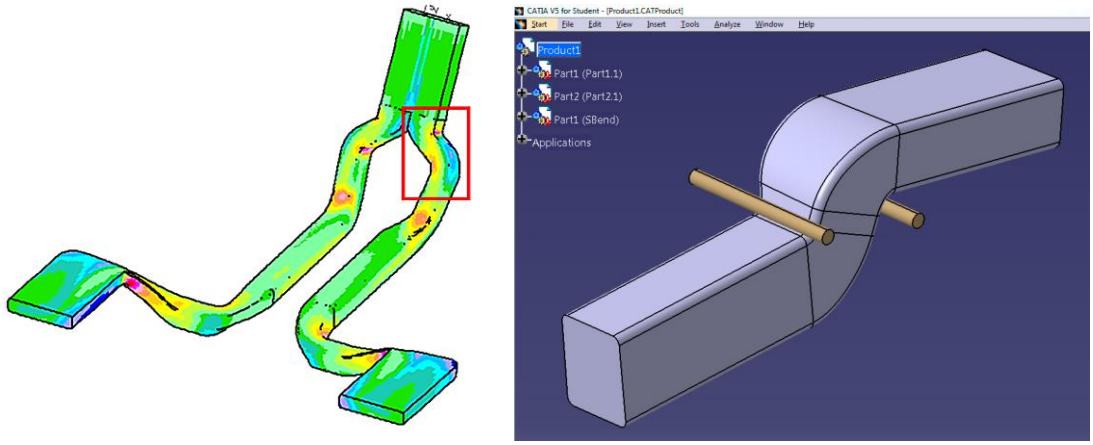


Figure 7.15 ONERA M6 wing optimized section at 60% span

7.4.4 S-Bend Optimization

The developed methodology is now demonstrated on a S-Bend automobile duct (discussed in Chapter 6) with multiple assembly constraints. Here the CAD model of the duct has 48 parameters (as optimized in Chapter 6) and was used to demonstrate the versatility of the developed framework to address scenario with multiple assembly constraints. The assembly components are created in CATIA V5 assembly workbench as shown in Figure 7.16, where two cylindrical components are used to represent different components in the assembly that constrain shape optimization of the S-Bend boundary. The location of constraints was selected such that they would restrict the shape change seen in Chapter 6. They were created such that in its initial state the two cylinders were adjacent to the S-Bend with clearance distances of 1.03 mm and 0.55 mm.



(a) (b)

Figure 7.16 (a) Automotive airduct [25], (b) Assembly of S-Bend with other components

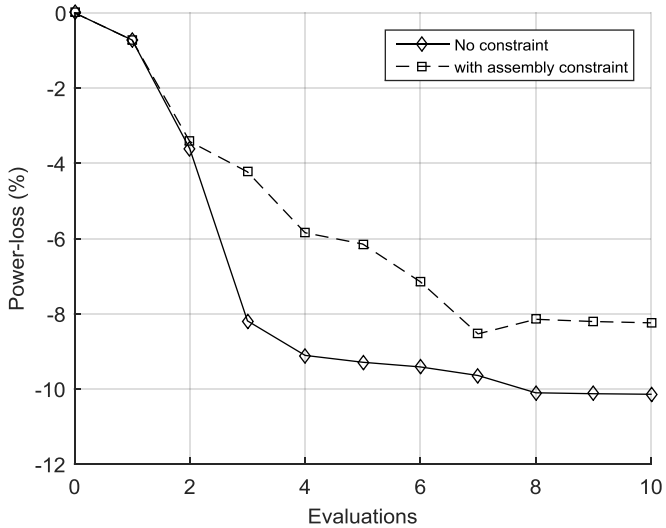


Figure 7.17 Optimization history for S-Bend with assembly constraint

The assembly configuration of the S-Bend after the optimization (10 steps) was found to be very close to contact with the two cylinders. A reduction in power-loss of 8.35% was achieved for the S-Bend, with the assembly constraints in place, compared to 10.14% achieved when optimized without any constraints. The optimization history for minimizing the objective function is shown in Figure 7.17. The optimization starts with minimization of the objective function, which also leads to reduction in the clearance between the S-Bend and assembly components. It should be noted that at iteration-7 of the optimization, the geometry is in clash with one of the parts in the product assembly, and to remove clash the optimizer moves the geometry such that an increase in objective function is observed.

7.5 Summary

The objective of this chapter was to present an efficient approach to CAD-based adjoint shape optimization with packaging constraints imposed by other components during the product assembly. In order to facilitate the computation of these constraints and their respective gradients, a series of CAD system APIs were developed in Python 3.5 to interact with the product assembly workbench of CATIA V5 to extract interference distances for different assembly components. In this Chapter, the models created in CATIA V5 were used as the assembly constraint, but the framework can also be used for models created in other CAD systems like SIEMENS NX, SolidWorks etc. to impose the assembly constraints.

The applicability and scope of the developed framework was demonstrated for three different test cases of varying complexity ($2D \rightarrow 3D \rightarrow 3D$ multiple components). The developed framework was firstly applied for the optimization of a simple beam model constrained by a 2D block in the assembly. For the aerodynamic test cases of NACA0012 aerofoil and ONERA M6 wing, the CFD analysis was performed using an open-source CFD solver SU2, and the assembly constraints were imposed using 2D and 3D fuel-box respectively. The versatility and applicability of the developments was extended and demonstrated for optimizing designs with multiple assembly components, where the S-Bend duct was optimized to minimize the power-loss in the presence of two cylindrical components restricting the movement of S-Bend portion of the duct.

Chapter 8 Discussion

8.1 Design velocity and gradient computation

The overall aim of this thesis was to facilitate the use of commercial CAD systems in an industrial optimization workflow. The first step in this direction was the development of a robust and efficient method for computing the parametric design velocity for the parameters defining the CAD model built using commercial CAD system, as outlined in Chapter 3.

The computation of design velocities is based on STEP files which facilitates its implementation to any parametric CAD model built using commercial CAD systems. The success of design velocity computations depended on the ability of facets to resolve the curvature in the original and perturbed models, and it was shown that GMSH can be effectively used to meet this requirement (at least for the test cases analysed). One of the areas for future-work would be to apply curvature sensitive meshing to each perturbed model and ensuring that the maximum mesh density required for a face in any of the perturbed models is reflected in the faceting of the original model.

Also, the applicability of the design velocities for the prediction of the performance gradients, required the design velocities to be computed accurately in the regions of high surface sensitivity (as given by the adjoint). It is difficult to gauge a priori where a need for the high density of elements is required. In this work, this was achieved by using a dense geometrical faceting in regions by first running the adjoint and assessing the results manually to determine where the need occurred. The density was then controlled by manually setting the surface mesh generators. Future work would be to explore the use of metrics based on the change in curvature caused by changes in the parameters, to adapt the mesh of the original model.

The proposed approach to compute the design velocity was unaffected by changes in the boundary topology of the CAD model, which hampers the majority of existing approaches of design velocity computations for CAD models, but which are likely to occur during shape optimization of complex models. This was observed for the NGV test case where the parametric perturbations led to the appearance of new sliver faces in the trailing edge slot region (Figure 8.1), while during the optimization of the ONERA M6 wing sliver faces appeared near to the leading edge of the wing where the surface curvature was high (Figure 8.2).

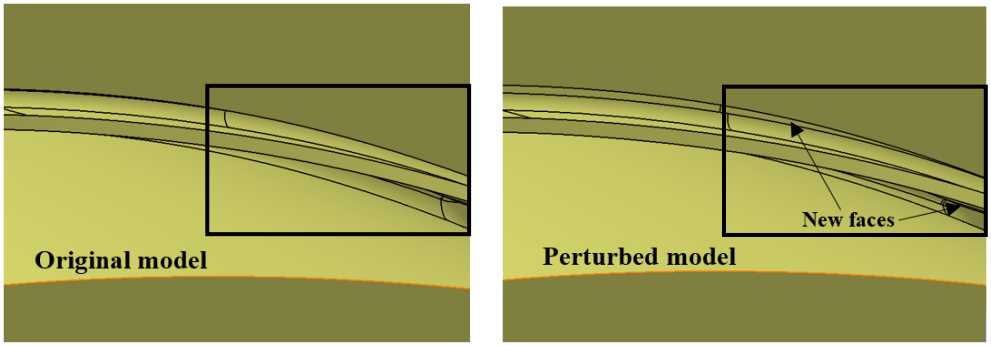


Figure 8.1 Parametric perturbation causing appearance of sliver face on NGV model

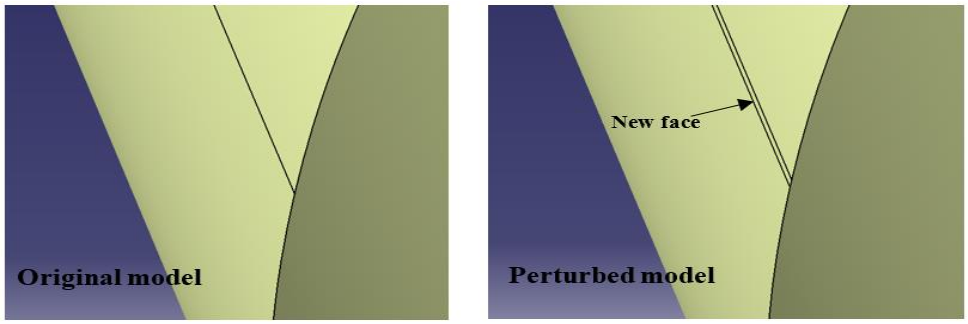


Figure 8.2 Parametric perturbation causing appearance of sliver face on ONERA wing

For each of the examples shown in chapter 3 and chapter 4, the cost of computing the design velocities for all parameters was small compared to the cost of computing the adjoint sensitivities. Also, it is proposed to do this step in parallel with the primal/adjoint computation, which means that the cost of including additional parameters would add no additional time to the overall optimization loop. That said, for models of very complex shapes defined by large numbers of parameters, there will be a point where the cost of computing the design velocities will become greater than the cost of the primal/adjoint computation as for the test case in section 5.4.4. At this point, to reduce the computational cost, it is possible to parallelize the process by distributing the calculation of design velocities for different parameter sets across

different computers. Doing so will require an additional CAD license for each additional machine used, however one additional license will half the time to compute design velocities for all parameters (and 3 will reduce it to a third etc). It is difficult to imagine a scenario where more than a small number of CAD licenses will be required. There is also unlikely to be a scenario where a company is working with a model of such complexity to require parallelisation across many machines, but yet does not have sufficient licenses to allow the parallelisation to take place in the time required to analyse the complex shape. It should also be noted that using parametric effectiveness to reduce the number of parameters considered in an optimization as described in Chapter 5 will also result in the reduction of the required time.

The objective of the work presented in Chapter 4 was to exploit the design velocity approach and develop an automated optimization framework (see Figure 4.1) linking CAD systems, meshing software, computational analysis codes and optimization algorithms. The versatility and applicability of the developed framework to deal with models built in commercial CAD system and analysed using state-of-the-art CFD solver was demonstrated, which substantiated its applicability to industrial CAE systems where models are generally built using commercial CAD systems and analyzed using different CFD codes. The advantage of using this approach lies in the fact that the optimization is performed directly on the CAD models, and consequently the optimized geometry is available in the CAD package and can be directly used for other design applications. Although, in Chapter 4 optimization of only aerodynamic test cases were presented, the process can be easily extended for the optimization of different categories of test cases. This was shown in chapter 5 and chapter 6, where the developed workflow is used for the optimization of test cases from structural mechanics and the automobile industry.

One of the limitations of the optimization framework presented here was that it required an automated methodology to create a new CFD mesh at each optimization step. For complex models, this can require a lot of effort and expertise on the part of the user even for simple test cases and can become in-feasible when optimizing larger test cases (e.g. full aircraft). So, as part of the future work it would be beneficial to link the optimization framework with a robust mesh deformation method which is not limited by the change in the boundary topology of the model, and which can efficiently use the CAD geometry (at each optimization step) to move the initial mesh to be used

for CFD analysis. Whilst using directly the parametric CAD model in the design optimization has the advantage that the optimized design is available as a CAD model and can be directly used for downstream processes without any post-processing. On contrary, this increases the workload on designers to ensure the quality of parametrization for the optimization.

8.2 Parametric effectiveness for efficient adjoint optimization

The motivation for the work presented in Chapter 5 was drawn from the fact that a CAD model can be parametrized in different ways and it is not obvious that which type of parameterization is best suited for the purpose of optimization. Although the theoretical work on parametric effectiveness was presented by Robinson et. al. [103], authors could not automate the approach to compute the effectiveness of different combinations of CAD parameterization. Contributing towards the overall aim of an efficient CAD-based optimization workflow, an automated framework was developed which can run on a fly to compute the parametric effectiveness of any combination of CAD parameters. Later, an automated approach was developed to select a smaller subset of parameters with highest parametric effectiveness.

For the automotive ventilation duct model obtained from Volkswagen Group Research, Germany (see Figure 5.2), it was interesting to find that the parametric effectiveness of all the CAD parameters was substantially lower ($= 0.08$). This could be justified from the Figure 5.3, where adjoint sensitivities suggested the movement of only a small part of the boundary, while using all the CAD parameters would have moved the very large part of the boundary (see Figure 5.4(a)). While analyzing the parameters selected by the designers, it was seen they moved the boundary precisely in the region of high adjoint sensitivity but had slightly lower parametric effectiveness ($= 0.47$) than that selected using the approach developed in Chapter 5 ($= 0.53$). Again, the parameters obtained using the automated process also contained the parameters selected by the designers. Thus, this approach of parameter selection could be efficiently configured to assist the designers in the selection of parameters to be used for optimization. This would help to workload of designers and analysts who currently use their engineering skills and requires a considerable amount of time.

In theory, when the optimization converges, all gradients are expected to get reduced to a very low value (~ 0), which signifies that the higher the number of design variables

(more degrees of freedom), the higher is the performance gain. So, it is a trade-off between the performance and computational requirement. Also, it was interesting to note that for the S-Bend test case, while using all the parameters for optimization the performance gain in the first 4 iterations surpassed the maximum performance achieved (for full optimization) by using the selected set of CAD parameters.

The application of the developed methodologies for the optimization of the DrivAer model, demonstrated that in some scenarios updating many parameters in a CAD model of high complexity at one time can be computationally much more expensive. In fact, each update required much more time than that required for the CFD analysis. For this test case, the CAD update at one optimization step was completed in approximately 3 hours (on a 3.60GHz workstation with 16GB RAM), while the primal and adjoint analysis took approximately 1 hour (on a high-performance cluster with 216 CPU cores) each for primal and adjoint analysis. In these situations, it becomes essential to select a reduced set of optimum parameters, and the obvious choice is to use directly the parametric sensitivities to select these parameters. Interestingly, it was found that the most effective parameter combination had significantly higher parametric effectiveness ($= 0.83$) than that obtained when the same number of parameters with high parametric sensitivity were analysed (parametric effectiveness $= 0.59$). It is to be noted that the most effective combination of parameters were not the ones with highest individual parametric effectiveness.

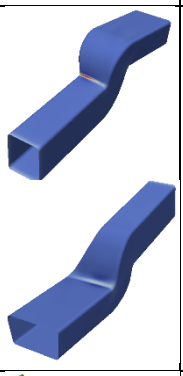
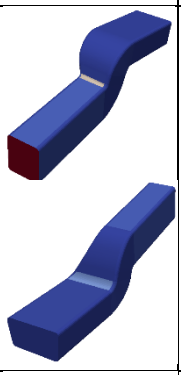
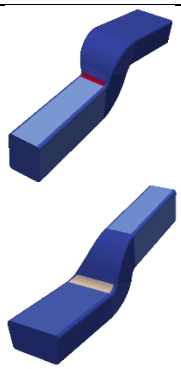
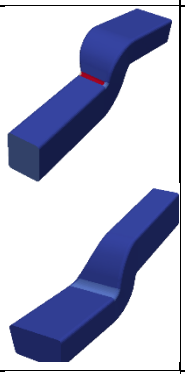
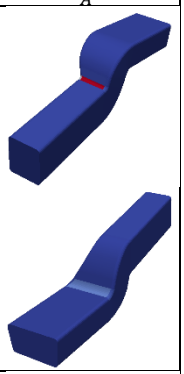
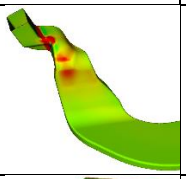
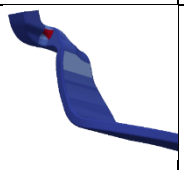
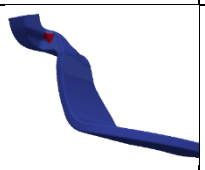
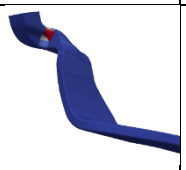
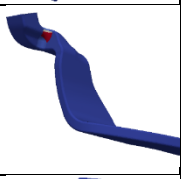
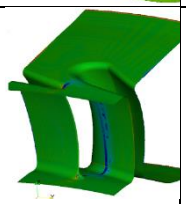
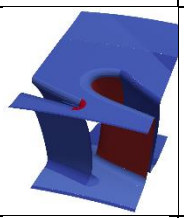
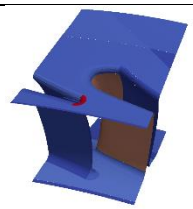
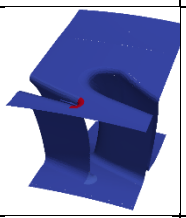
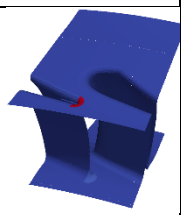
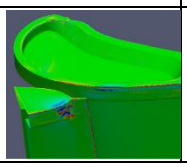
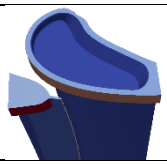
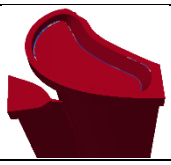
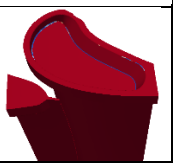
One of the limitations of the approach of the parametric effectiveness computations is that it requires the computation of design velocities for all the CAD parameters for the first step. This can be computationally very expensive as it requires the creation of STEP files for each parametric perturbation and subsequently generate surface facets for each STEP file. For the DrivAer test case, this step took approximately 26 hours on 3.60GHz workstation. This is one example where to reduce the computational time (required to compute the design velocities) would be to parallelize the work among different workstations and only link the results together while computing the parametric effectiveness for down-selecting the parameters.

8.3 Automatic refinement of CAD parameterization

The objective of Chapter 6 was to present an efficient methodology to enhance the flexibility of the CAD model by intelligently inserting optimum new features into the

CAD model feature tree. The approach described was dependent on the geometry on which the geometrical feature was being inserted to, but this could be further improved potentially by associating adjoint surface sensitivities with different faces in the CAD model. Table 8.1 shows analysis of some parameters (based on adjoint sensitivity) that can be used to automatically select the regions where the addition of new features would be beneficial, including

Table 8.1 Analysis of adjoint sensitivities for feature insertion

Model	Sensitivity (ϕ)	Measure 1 $\int \phi^2 dA$	Measure 2 $\int (\phi - \bar{\phi})^2 dA$	Measure 3 $\frac{\int \phi^2 dA}{A}$	Measure 4 $\frac{\int (\phi-\bar{\phi})^2 dA}{A}$
S-Bend duct					
Ventilation duct					
NGV					
Turbine rotor blade					

Measure 1:$\int \phi^2 dA$ gives the expected change in objective function if each face of the CAD model is moved according to the adjoint sensitivities.

Measure 2:$\int (\phi - \bar{\phi})^2 dA$ gives the variation in adjoint sensitivity over a face. This value would be zero if the sensitivity is uniform over the face and would suggest moving the face uniformly either inward or outward.

Measure 3: $\frac{\int \phi^2 dA}{A}$ gives the change in objective function per unit area. This parameter can be biased for faces with larger areas, as it will smear out the sensitivities.

Measure 4: $\frac{\int (\phi - \overline{\phi})^2 dA}{A}$ gives the variation in sensitivity over a face with respect to its area. This value will be higher for the faces with uniform sensitivities but lower area. So, if there are two faces with small variation in sensitivities then this parameter will give preference to face with a lower area.

For some initial tests, the last measures 3 and 4 completely fail for turbine rotor blade, where several small area faces existed around the filleted corners and would result in higher values for these measures. For the NGV model, though the first two measures indicate similar faces to be moved, but 2nd measures prefers face with a smaller area. For S-Bend a combination of measures 1 and 2 can be effectively used to reach the same faces to be moved. Since the sensitivities near to the inlet are generally high (due to flow fluctuations), so measure 1 indicated the movement of faces near to inlet in addition to faces near the bends. For the ventilation duct, the measures were calculated by removing the inlet and outlet extrusions. In this case, all the 4 measures indicated the same faces to be moved in the high sensitivity regions. Using these measures, it was difficult to reach a conclusive inference which may result in a general automation methodology.

In Chapter 6, methodologies were formulated to refine the existing CAD parameterization and insert new shape-features based on the original CAD entity or the feature type. The methodologies used were: i) replace a straight line or circle entities in a sketch with splines, ii) increase the number of control points defining a spline, which provides additional shape flexibility and iii) to insert a new profile along a multi-section solid such that there is more control over the shape of a body along its length. This step resulted in increase in the number of CAD design parameters, and the effect of newly added parameters was quantified using adjoint sensitivities. In Chapter 6, the insertion of CAD features was performed for CAD models created in CATIA V5, extension of the developed algorithms and processes to other CAD systems would be a part of the future work.

It should be noted that the approaches used may sometimes result in the features getting inserted where parameters already exist, or where more design flexibility is not required. But these situations can be identified by a decrease in the value of parametric effectiveness when such features are inserted, as increasing number of parameters moving the same region of the boundary (or moving less sensitive regions) will be

penalised by the unit-2 norm constraint inherited in the calculation of parametric effectiveness.

This work of adding flexibility to the CAD model can be compared with the works of Weiss [126], where the author gradually increased the shape complexity of spline structures by varying the number of control points during optimization. The approach imparted flexibility to the design but required access to the expensive CAA RADE environment to alter the internal CATIA V5 data, while no such alteration of the internal CAD data was made in this work. In this thesis, the insertion of new CAD features was done directly into the CAD feature tree using the Python CAD system APIs.

The Python Scipy optimization framework used in chapter 6 was configured to automatically insert new CAD features either at the start of optimization or during the optimization. For the S-Bend test-case, the comparison of optimization results for the initial CAD parameters, and the ones created automatically by inserting new feature into the CAD model, showed that the CAD model with highest parametric effectiveness resulted in a higher reduction in objective function compared to the ones with lower parametric effectiveness. Analysing the optimization results in Figure 6.15, it was seen that optimization with the newly inserted features (in high adjoint sensitivity regions), the performance gain in first 5 iterations surpassed the performance gains obtained using the original parameters in 45 iterations. Also, new features insertion resulted in more than 7% reduction in power-loss compared to that achieved with the original parameters.

Further, analysing the computational time required for the optimization processes, it was found that the optimizing design with newly added features took more time compared to optimization using original parameters. This increased computational time resulted from the increase in the number of iterations taken by the optimizer for convergence (54 iterations instead of 42). The optimized S-Bend duct showed convex regions (ripples) at the intersection of S-Bend area with the fixed inlet and outlet duct, which is non-intuitive and required further analysis of the flow field as shown in Figure 6.19 and Figure 6.20.

The addition of new features in the S-Bend was done particularly in the areas of high adjoint sensitivities, so the computed gradients with respect to these newly created

parameters was high resulting in large geometrical movement in these areas. Also, the inlet and outlet duct were constrained to move, so deformations get localised and led to the formation of ripples. It is expected that a small re-circulation region forms in these ripple areas resulting in a low adjoint sensitivity. Owing to this, once the ripples are formed the geometry is unable to deform to a smooth shape. Although, the ripples in the geometry are not appeasing features but they also do not cause any harm regarding the optimum.

An interesting application of the design velocities combined with feature insertion methodology was demonstrated for fitting a feature-based CAD model to a mesh. The approach to define the objective function to minimize the distance between a CAD model and a mesh, showed the prospect of applying the developed methodologies for post-processing and fit a feature-based CAD directly from the optimized mesh. The main benefit of this method is that it allows the exploitation of a large design space in terms of flow optimization, and at the same time facilitates the availability of CAD geometry which can be used for downstream applications.

8.4 CAD-based adjoint optimization with assembly constraints

The objective of Chapter 7 was to develop an efficient methodology for shape optimization using the assembly constraints imposed by the other components during the product assembly. In this regard the capabilities of the modern CAD system DMU was exploited. A series of CAD system APIs developed for the purpose were able to use the interference detection system in CATIA V5 product assembly to compute the interferences between different assembly components.

The developed framework was first applied to the optimization of a simple beam model (analyzed in ABAQUS) constrained by a 2D block in the assembly. The objective function used was minimization of strain energy of the system which was a self-adjoint problem and thus required only one analysis to provide the surface sensitivities. It was interesting to note that for the unconstrained optimization results in Figure 7.4(a), the strain energy density in the entire model was of the same colour. This indicates that for this model there is no further performance improvement possible (without removing the constraint of constant volume).

The objective of NACA0012 aerofoil and ONERAM6 wing optimization was to minimize the aerodynamic drag in the presence of fuel-box inside the model. In a

broader perspective, the fuel-box constrain could act as a representative of the regions inside the wing which can be used for storing structural equipment, landing gear etc. The objective of the S-Bend duct optimization was to minimize the power-loss in the duct in the presence of two representative cylindrical components restricting the movement of the duct. These constraints are representative of the actual constraints imposed by the steering column and other mechanical equipment.

The developed optimization framework successfully optimized the component without introducing interference during the optimization. It was interesting to note that for all the examples, optimizing the models without considering adjacent components, resulted in optimized shapes which would have caused fit issues when assembly would have been attempted.

Since, the optimization algorithm SLSQP uses a least-square formulation to compute the Lagrangian multipliers to incorporate the constraints in the optimization, the efficiency of this constrained optimization framework also depends on the number and types of constraints. Within the scope of this work, only one or two adjacent components were used as the assembly constraints, and the applicability on a complex industrial component with comparatively higher number of assembly components is yet to be demonstrated. Here, the complexity arises from the perspective of optimizer but not from the methodology described in Chapter 7 for the computations of assembly constraint and their respective gradients. Thereafter, one of the other limitations of this approach is the requirement of a suitable scaling factor which is important to consider when dealing with non-linear optimization packages. Here, the objective function is based on the CFD analysis, while the constraint is based on the clearance distance between the two components, so a considerable effort is required to choose a suitable scaling factor for the objective as well as the constraints.

Chapter 9 Conclusions

In this thesis, an automated optimization process was presented, which uses the parameters defining the features in a feature-based CAD model as design variables. It was built on previous works on design velocity, parametric effectiveness and CAD feature insertion. Novel methods have been used to develop a more robust and efficient tool to calculate design velocity than the one presently existed. This contributed substantially towards the development of an automated optimization framework linking commercial CAD systems and analysis software. The main contributions of this research are summarized below:

- With the developments in Chapter 3, an efficient and robust approach is formulated which can compute the design velocities for parametric CAD models build in commercial CAD systems without any user intervention.
- Using the automated optimization framework in Chapter 4, it is now possible to optimize directly the parametric models build in commercial CAD systems and analysed using in-house or commercial CFD software.
- An automated framework was developed in Chapter 5, which enables the computation of the parametric effectiveness of different CAD parameters and can act as an efficient tool to rate the quality of different types of parametrization schemes.
- The automated tool developed to select the optimum CAD parameters, could be used to remove the common roadblock for optimizing industrial components, which are mainly designed using a large number of parameters.
- The Novel methodologies developed in Chapter 6 for inserting new CAD features can automatically increase the design freedom of the model. This can

be used to obtain a better optimized component than would have been possible using the initial parameters alone.

- An automated optimization framework was developed in Chapter 7 which can incorporate constraints imposed by the adjacent components in the CAD product assembly within the optimization.

Chapter 10 Future works

Some of the work presented in this research need to be further investigated. A general optimization framework utilizing the CAD models created in a feature-based CDA systems have been developed in this thesis, still there are possibilities of extension and improvements.

- The methodologies developed in this thesis used adjoint sensitivities, which limits their applicability for optimizing designs for a given objective function. Exploiting these methods for performing a multi-disciplinary design optimization by computing multi-disciplinary adjoint sensitivities (structural and aerodynamics) will be an area for future research.
- The calculation of design velocity in Chapter 3 required an efficient and robust method for creating surface mesh. The current approach used background mesh method in GMSH which can be enhanced further. An interesting area of research would be to analyse how the curvature in each of the perturbed models can be used to drive a metric-based mesh refinement process for the mesh of the original model.
- The optimization framework presented in this thesis utilizes a remeshing process to create the analysis mesh at each optimization step. As part of future research, it would be interesting to link the developed approaches with an efficient mesh deformation tool which is robust under a change in the boundary topology of the CAD model.
- In Chapter 5, the automated framework was presented to select the optimum combination of parameters to be used for optimization. The applicability of this approach for a range of test cases would be required before a general conclusion can be made about the advantages of reducing the design space for

a gradient based adjoint optimization. Moreover, for global optimization algorithms where only a limited number of design parameters can be used, this approach might be beneficial (as a pre-processing step) to select the most appropriate parameters for the optimization.

- In Chapter 6, the increase in the parametric effectiveness obtained by inserting new CAD features used as the rationale to decide which new CAD features to insert. It would be interesting to devise an automated methodology to use adjoint surface sensitivity to automatically guide the insertion of new CAD features. Moreover, the approaches were developed to automatically insert optimum CAD features which can modify geometrical shapes, without changing the model's topology. Extension of these methodologies to add features causing topological changes to the model (like vortex generators) are to be considered for future research.
- The optimization framework presented in Chapter 7, utilized the prior information about other components in the product assembly to define optimization constraints. In future, the applicability of the developed framework is to be tested for the optimization of an industrial component in the presence of significantly larger number of other components in the product assembly.

References

1. CATIA V5. http://www.3ds.com/products-services/catia/. Accessed 01/27/2017
2. Siemens NX. https://www.plm.automation.siemens.com/en/products/nx/about-nx-software.shtml. Accessed 10/10/2017
3. SolidWorks. http://www.solidworks.co.uk/sw/products/3d-cad/packages.htm. Accessed 01/22/2018
4. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning, 1989. Reading: Addison-Wesley
5. Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation 3:1-16. doi: https://doi.org/10.1162/evco.1995.3.1.1
6. Shahpar S (2011) Challenges to overcome for routine usage of automatic optimisation in the propulsion industry. Aeronaut J 115:615-625. doi: https://doi.org/10.1017/S0001924000006308
7. Robinson TT, Armstrong CG, Chua HS, Othmer C, Grahs T (2012) Optimizing parametrised CAD geometries using sensitivities based on adjoint functions. Computer Aided Design and Application 9:253-268. doi: https://doi.org/10.3722/cadaps.2012.253-268
8. Giles MB, Pierce N (2000) An introduction to the adjoint approach to design. Flow Turbul Combust 65:393-415. doi: https://doi.org/10.1023/A:1011430410075
9. Giles MB, Duta MC, Müller J, Pierce N (2003) Algorithm developments for discrete adjoint methods. AIAA J. 41:198-205. doi: https://doi.org/10.2514/2.1961
10. Jameson A (2003) Aerodynamic shape optimization using the adjoint method. Lectures at the Von Karman Institute, Brussels
11. Reuther J, Alonso JJ, Rimlinger MJ, Jameson A (1999) Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. Computer and Fluids 28:675-700. doi: https://doi.org/10.1016/S0045-7930(98)00050-4

12. Brezillon J, Gauger N (2004) 2D and 3D aerodynamic shape optimization using the adjoint approach. Aerosp Sci Technol 8:715-727. doi: https://doi.org/10.1016/j.ast.2004.07.006
13. Anderson WK, Venkatakrishnan V (1999) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. Comput Fluids 28:443-480. doi: https://doi.org/10.1016/S0045-7930(98)00041-3
14. Mader CA, RA Martins J, Alonso JJ, Der Weide EV (2008) Adjoint: an approach for the rapid development of discrete adjoint solvers. AIAA J 46:863-873. doi: https://doi.org/10.2514/1.29123
15. Roth R, Ulbrich S (2013) A discrete adjoint approach for the optimization of unsteady turbulent flows. Flow Turbul Combust 90:763-783. doi: https://doi.org/10.1007/s10494-012-9439-3
16. Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. Int J Numer Methods Fluids 58:861-877. doi: https://doi.org/10.1002/fld.1770
17. Chang K, Silva J, Bryant I (1999) Concurrent design and manufacturing for mechanical systems. Concurrent Engineering 7:290-308. doi: https://doi.org/10.1177/1063293X9900700403
18. IODA. http://ioda.sems.qmul.ac.uk/. Accessed 10/04/2017
19. FlowHead. http://flowhead.sems.qmul.ac.uk/. Accessed 10/04/2017
20. AboutFlow. http://aboutflow.sems.qmul.ac.uk/. Accessed 10/04/2017
21. Zingg DW, Nemec M, Pulliam TH (2008) A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 17:103-126.
22. Lombardi G, Mengali G, Beux F (2006) A hybrid genetic based optimization procedure for aircraft conceptual analysis. Optimization and Engineering 7:151-171. doi: https://doi.org/10.1007/s11081-006-6837-1
23. Kelner V, Grondin G, Léonard O, Moreau S (2005) Multi-objective optimization of a fan blade by coupling a genetic algorithm and a parametric flow solver. In: 6th International Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems (EUROGEN 2005). Munich

24. Keane Andy, Nair P (2005) Computational approaches for aerospace design: the pursuit of excellence. John Wiley & Sons,
25. Othmer C (2014) Adjoint methods for car aerodynamics. Journal of Mathematics in Industry 4:1-23. doi: https://doi.org/10.1186/2190-5983-4-6
26. Kammerer S, Mayer J, Paffrath M, Wever U, Jung A (2003) Three-dimensional optimization of turbomachinery bladings using sensitivity analysis. ASME Turbo Expo: Power for Land, Sea, and Air 6:1093-1101. doi: https://doi.org/10.1115/GT2003-38037
27. Papadimitriou D, Giannakoglou K (2007) Total pressure loss minimization in turbomachinery cascades using a new continuous adjoint formulation. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 221:865-872. doi: https://doi.org/10.1243/09576509JPE463
28. Walther B, Nadarajah SK (2015) Adjoint-based constrained aerodynamic shape optimization for multistage turbomachines. J Propul Power 31:1298-1319. doi: https://doi.org/10.2514/1.B35433
29. Othmer C, Grahs T (2005) Approaches to fluid dynamic optimization in the car development process. In: International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control With Applications to Industrial and Societal Problems
30. Thomas S, Carsten O (2016) Adjoint optimization for vehicle external aerodynamics. International Journal of Automotive Engineering 7:1-7. doi: https://doi.org/10.20485/jsaeijae.7.1_1
31. Adjoint-Based Topology and Shape Optimization for Car Development. https://www.nas.nasa.gov/assets/pdf/ams/2014/AMS_20141202_Othmer.pdf. Accessed 10/12/2017
32. Shape Optimization for Aerodynamic Efficiency Using Adjoint Methods. http://www.cadfemukandireland.com/wp-content/uploads/2016/09/wp-shape-optimization-for-aero-efficiency-adjoint-methods.pdf. Accessed 10/12/2017
33. Pironneau O (1974) On optimum design in fluid mechanics. J.Fluid Mech. 64:97-110. doi: https://doi.org/10.1017/S0022112074002023
34. Castro C, Lozano C, Palacios F, Zuazua E (2007) Systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. AIAA J 45:2125-2139. doi: https://doi.org/10.2514/1.24859

35. Papadimitriou DI, Giannakoglou KC (2006) Compressor blade optimization using a continuous adjoint formulation. In: ASME Turbo Expo 2006: 1309-1317.https://doi.org/10.1115/GT2006-90466
36. Nielsen EJ, Anderson WK (1999) Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. AIAA J 37:1411-1419. doi: https://doi.org/10.2514/2.640
37. Nadarajah SK, Jameson A (2000) A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In: 38th Aerospace Sciences Meeting and Exhibit: 667.https://doi.org/10.2514/6.2000-667
38. OPENFOAM. http://www.openfoam.com/. Accessed 09/27/2017
39. Economon TD, Palacios F, Copeland SR, Lukaczyk TW, Alonso JJ (2016) SU2: An open-source suite for multiphysics simulation and design. AIAA J 54:828-846. doi: https://doi.org/10.2514/1.J053813
40. HELYX. http://engys.com/products/helyx. Accessed 01/27/2017
41. Schwamborn D, Gerhold T, Heinrich R (2006) The DLR TAU-code: recent applications in research and industry. In: ECCOMAS CFD 2006: proceedings of the european conference on computational fluid dynamics, The Netherlands, 5-8 September, 2006
42. HYDRA. https://www.mpls.ox.ac.uk/research/the-hydra-code-rolls-royces-standard-aerodynamic-design-tool. Accessed 01/27/
43. Samareh J (2001) Survey of shape parametrisation techniques for high-fidelity multidisciplinary shape optimization. AIAA J 39:877-884. doi: https://doi.org/10.2514/2.1391
44. Belegundu A, Rajan S (1988) A shape optimization approach based on natural design variables and shape functions. Comput.Methods Appl.Mech.Eng. 66:87-106. doi: https://doi.org/10.1016/0045-7825(88)90061-8
45. Jameson A, Martinelli L, Pierce N (1998) Optimum aerodynamic design using the Navier–Stokes equations. Theor. Comput. Fluid Dyn. 10:213-237.
46. Helgason E, Krajnovic S (2012) Aerodynamic shape optimization of a pipe using the adjoint method. In: ASME International Mechanical Engineering Congress & Exposition, Nov: 9-15.
47. Jameson A (2004) Efficient aerodynamic shape optimization. AIAA paper 4369:2004.

48. Samareh J (2004) Aerodynamic shape optimization based on free-form deformation. In: 10th AIAA Multidisciplinary analysis and optimization conferencehttps://doi.org/10.2514/6.2004-4630
49. Kenway G, Kennedy GJ, Martins J (2010) A CAD-free approach to high-fidelity aerostructural optimization. In: Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX
50. Hoogervorst JEK, Elham A (2016) Wing aerostructural optimization using the individual discipline feasible architecture. In: 17th AIAA/ISSMO multidisciplinary analysis and optimization conference: 3996.
51. Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. In: 13th annual conference on computer graphics and interactive techniques 20: 151-160.
52. Anderson CW, Crawford-Hines S (2000) Fast generation of nurbs surfaces from polygonal mesh models of human anatomy. Colorado State University Computer Science Technical Report CS-99 101
53. Guo B (1997) Surface reconstruction: from points to splines. Comput.-Aided Des. 29:269-277. doi: https://doi.org/10.1016/S0010-4485(96)00055-3
54. Martin MJ, Andrés E, Widhalm M, Bitrián P, Lozano C (2012) Non-uniform rational B-splines-based aerodynamic shape design optimization with the DLR TAU code. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 226:1225-1242. doi: https://doi.org/10.1177%2F0954410011421704
55. Yu G, Müller J, Jones D (2011) CAD based shape optimization using adjoint sensitivities. Comput Fluids 46:512-516. doi: https://doi.org/10.1016/j.compfluid.2011.01.043
56. Xu S, Jahn W, Müller J (2014) CAD based shape optimization with CFD using a discrete adjoint. Int J Numer methods Fluids 74:153-168. doi: https://doi.org/10.1002/fld.3844
57. Xu S, Timme S, Mykhaskiv O, Müller J (2017) Wing-body junction optimisation with CAD-based parametrisation including a moving intersection. Aerospace Science and Technology 68:543-551. doi: https://doi.org/10.1016/j.ast.2017.06.014
58. Burgos MJ (2015) NURBS-Based Geometry Parameterization for Aerodynamic Shape Optimization. Dissertation, Telecomunicacion

59. Xu S, Radford D, Meyer M, Müller J (2015) CAD-based adjoint shape optimisation of a one-stage turbine with geometric constraints. In: ASME Turbo Expo GT2015-42237https://doi.org/10.1115/GT2015-42237
60. Jesudasan R, Zhang X, Mueller J (2017) Adjoint optimisation of internal turbine cooling channel using NURBS-based automatic and adaptive parametrisation method. In: ASME Gas Turbine Conference, GTIndia2017-4669
61. Nurdin A, Bressloff NW, Keane AJ, Holden CME (2012) Shape optimisation using CAD linked free-form deformation. The Aeronautical Journal (1968) 116:915-939. doi: 10.1017/S0001924000007351
62. Koch J, Papoutsis-Kiachagias E, Giannakoglou K (2017) Transition from adjoint level set topology to shape optimization for 2D fluid mechanics. Comput.Fluids 150:123-138. doi: https://doi.org/10.1016/j.compfluid.2017.04.001
63. Shah JJ (1991) Assessment of features technology. Comput.-Aided Des. 23:331-343. doi: https://doi.org/10.1016/0010-4485(91)90027-T
64. Latif M, Boyd R, Hannam R (1993) Integrating CAD and manufacturing intelligence through features and objects. International Journal of Computer Integrated Manufacturing 6:87-93. doi: https://doi.org/10.1080/0951192930894455 8
65. Salomons OW, van Houten FJ, Kals H (1993) Review of research in feature-based design. Journal of Manufacturing Systems 12:113-132. doi: https://doi.org/10.1016/0278-6125(93)90012-I
66. Chen S, Torterelli D (1997) Three-dimensional shape optimization with variational geometry. Structural Optimization 13:81-94. doi: https://doi.org/10.1007/BF01199226
67. Truong AH, Zingg DW, Haimes R (2016) Surface mesh movement algorithm for computer-aided-design-based aerodynamic shape optimization. AIAA J 54:542-556. doi: https://doi.org/10.2514/1.J054295
68. Hardee E, Chang K, Tu J, Choi KK, Grindeanu I, Yu X (1999) A CAD-based design parameterization for shape optimization of elastic solids. Adv Eng Softw 30:185-199. doi: https://doi.org/10.1016/S0965-9978(98)00065-9
69. Chang K, Choi KK, Tsai C, Chen C, Choi BS, Yu X (1995) Design sensitivity analysis and optimization tool (DSO) for shape design application. Computing Systems in Engineering 6:151-175. doi: https://doi.org/10.1016/0956-0521(95)00006-L

70. Kripac J (1997) A mechanism for persistently naming topological entities in history-based parametric solid models. Comput.-Aided Des. 29:113-122. doi: https://doi.org/10.1145/218013.218024
71. Raghothama S, Shapiro V (1998) Boundary representation deformation in parametric solid modelling. 17:259-286. doi: https://doi.org/10.1145/293145.293148
72. Nemec M, Aftosmis M (2008) Adjoint sensitivity computations for an embedded boundary cartesian mesh methods. J Comput Phys 227:2724-2742. doi: https://doi.org/10.1016/j.jcp.2007.11.018
73. TAPENADE. https://www-sop.inria.fr/tropics/tapenade.html. Accessed 02/21/2018
74. Torreguitart IS, Verstraete T, Mueller L (2017) Optimization Of The Ls89 Axial Turbine Profile Using A Cad And Adjoint Based Approach. In: 12 th European Conference on Turbomachinery Fluid dynamics & Thermodynamics
75. Walther A, Griewank A (2010) A Package for the Automatic Differentiation of Algorithms Written in C/C. URL https://projects.coin-or.org/ADOL-C
76. Banovic M, Mykhaskiv O, Auriemma S, Legrand H, Walther A, Müller J (2017) Automatic Differentiation of the Open CASCADE Technology CAD System and its coupling with an Adjoint CFD. Optimization Methods & Software. doi: http://www.optimization-online.org/DB_HTML/2017/03/5931.html
77. Auriemma S, Banovic M, Mykhaskiv O, Legrand H, Müller J, Walther A (2016) Optimisation of a U-bend using CAD-based adjoint method with differentiated CAD kernel. In: ECCOMAS Congress
78. Thompson P (2014) Efficient CAD based Aerodynamic Design Optimization with Adjoint CFD (thesis).
79. STEP. http://www.steptools.com/stds/step/. Accessed 08/18/2017
80. CADfix. https://www.iti-global.com/cadfix. Accessed 09/27/2017
81. Shahpar S, Caloni S (2013) Aerodynamic optimization of high-pressure turbines for lean-burn combustion system. Journal of Engineering for Gas Turbines and Power 135:055001. doi: https://doi.org/10.1115/1.4007977
82. Hogner L, Meyer M, Nasuf A, Voigt P, Voigt M, Vogeler K, Berridge C, Goenaga F (2016) Analysis of high pressure turbine nozzle guide vanes considering geometric variations. In: ASME Turbo Expo GT2016-57502

83. Moinier P, Muller J, Giles MB (2002) Edge-based multigrid and preconditioning for hybrid grids. AIAA J. 40:1954-1960.
84. Abaqus CAE. https://www.3ds.com/products-services/simulia/products/abaqus/abaquscae/. Accessed 10/04/2017
85. Geuzaine C, Remacle JF (2009) GMSH: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer methods Eng 79:1309-1331. doi: https://doi.org/10.1002/nme.2579
86. Ericson C. (2005) Real-time collision detection. Morgan Kaufmann Publishers,
87. Friedmann J, Bentley JL, Finkel A (1977) An algorithm for finding best matches in logarithmic expected time. ACM Trans Math Softw 3:209-226. doi: https://doi.org/10.1145/355744.355745
88. Van den Braembussche R (1990) Subsonic and transonic blade design by means of analysis codes.
89. Verstraete T (2010) CADO: a computer aided design and optimization tool for turbomachinery applications. In: 2nd Int. Conf. on Engineering Optimization, Lisbon, Portugal, September: 6-9.
90. Vasilopoulos I, Agarwal D, Meyer M, Robinson TT, Armstrong CG (2016) Linking parametric cad with adjoint surface sensitivities. In: ECCOMAS Congress 2016 - proceedings of the 7th European congress on computational methods in applied sciences and engineering 2: 3812-3827.
91. Schabowski Z, Hodson H (2014) The reduction of over tip leakage loss in unshrouded axial turbines using winglets and squealers. Journal of Turbomachinery 136:041001. doi: https://doi.org/10.1115/1.4024677
92. iSIGHT. https://www.3ds.com/products-services/simulia/products/isight-simulia-execution-engine/. Accessed 09/07/2017
93. BOXER. http://www.cambridgeflowsolutions.com/en/products/boxer-mesh/. Accessed 01/27/2017
94. Scipy Optimize. https://docs.scipy.org/doc/scipy/reference/optimize.html. Accessed 01/27/2017
95. Dropbox Cloud. https://www.dropbox.com/en_GB/. Accessed 04/20/2018
96. Vassberg JC, Harrison NA, Roman DL, Jameson A (2011) A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem. In: 29th AIAA Applied Aerodynamics Conference, Honolulu, HIhttps://doi.org/10.2514/6.2011-3176

97. Aerodynamic Design Optimization: Drag Minimization of the NACA 0012 in Transonic Inviscid Flow. https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/Test%20Cases/ADODG%20Case%201%20and%202%20NACA0012%20and%20RAE%202822.pdf. Accessed 10/03/2017

98. Ansys ICEM CFD. http://www.ansys.com/en-GB/Services/training-center/platform/introduction-to-ansys-icem-cfd-Hexa. Accessed 09/27/2017

99. ONERA M6. https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing.html. Accessed 01/31/2018

100. Lyu Z, Kenway GK, Paige C, Martins J (2013) Automatic differentiation adjoint of the Reynolds-averaged Navier–Stokes equations with a turbulence model. In: 21st AIAA computational fluid dynamics conference 10: 2581.

101. Hewitt P, Marques S, Robinson TT, Agarwal D (2016) Aerodynamic optimization using Adjoint methods and parametric CAD models. In: ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering

102. Vandenberg B, Gooden J (1994) Low-speed surface pressure and boundary layer measurement data for the NLR 7301 airfoil section with trailing edge flap. In AGARD, A Selection of Experimental Test Cases for the Validation of CFD Codes, 2

103. Robinson TT, Armstrong CG, Chua HS (2013) Determining the parametric effectiveness of a CAD model. Engineering with Computers 29:111-126. doi: https://doi.org/10.1007/s00366-011-0248-4

104. Kefi S, Joneja A, KT Tse T, Li S (2018) Channel geometry optimization for vertical axis wind turbines in skyscrapers. Computer-Aided Design and Applications 15:211-218. doi: https://doi.org/10.1080/16864360.2017.1375671

105. Karpouzas GK, Papoutsis-Kiachagias EM, Schumacher T, Villiers Ed, Giannakoglou KC, Othmer C (2016) Adjoint optimization for vehicle external aerodynamics. International Journal of Automotive Engineering 7:1-7. doi: https://doi.org/10.20485/jsaeijae.7.1_1

106. Papoutsis-Kiachagias E, Magoulas N, Müller J, Othmer C, Giannakoglou KC (2015) Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. Comput. Fluids 122:223-232. doi: https://doi.org/10.1016/j.compfluid.2015.09.002

107. Heft AI, Indinger T, Adams N (2012) Experimental and numerical investigation of the DrivAer model. In: ASME 2012 Fluids Engineering Summer Meeting, Rio Grande, Puerto Rico 1: 41-51.https://doi.org/10.1115/FEDSM2012-72272
108. Papoutsis-Kiachagias E, Giannakoglou KC (2016) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. Archives of Computational Methods in Engineering 23:255-299. doi: https://doi.org/10.1007/s11831-014-9141-9
109. Kapellos C, Alexias P, Villiers Ed (2016) *The adjoint method for automotive optimization using a sphericity based morpher*. In: NAFEMS adjoint CFD Seminar 2016
110. Hammond Mark, Robinson A (2000) Python programming on win32: Help for windows programmers. O'Reilly Media Inc.,
111. Mei Y, Wang X, Cheng G (2008) A feature-based topological optimization for structure design. Advances in Engineering Software 39:71-87. doi: http://dx.doi.org/10.1016/j.advengsoft.2007.01.023
112. Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Computer Methods in Applied Mechanics and Engineering 192:227-246. doi: http://dx.doi.org/10.1016/S0045-7825(02)00559-5
113. Bui-Thanh T, Damodaran M, Willcox K (2004) Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. AIAA J. 42:1505-1516.
114. Obayashi S, Takanashi S (1996) Genetic optimization of target pressure distributions for inverse design methods. AIAA J. 34:881-886.
115. Giles MB, Drela M (1987) Two-dimensional transonic aerodynamic design method. AIAA J. 25:1199-1206. doi: https://doi.org/10.2514/3.9768
116. Borggaard J, Burns J (1997) A PDE sensitivity equation method for optimal aerodynamic design. Journal of Computational Physics 136:366-384.
117. Martin MJ, Andres E, Lozano C, Valero E (2014) Volumetric b-splines shape parametrization for aerodynamic shape design. Aerospace Science and Technology 37:26-36. doi: https://doi.org/10.1016/j.ast.2014.05.003
118. Mader CA, Martins JR (2013) Stability-constrained aerodynamic shape optimization of flying wings. J.Aircr.

119. Walther B, Nadarajah S (2013) Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. Journal of turbomachinery 135:021017. doi: https://doi.org/10.1115/1.4007502

120. Kontoleontos E, Papoutsis-Kiachagias E, Zymaris A, Papadimitriou D, Giannakoglou K (2013) Adjoint-based constrained topology optimization for viscous flows, including heat transfer. Engineering Optimization 45:941-961. doi: https://doi.org/10.1080/0305215X.2012.717074

121. Boyse JW (1979) Interference detection among solids and surfaces. Commun ACM 22:3-9. doi: https://doi.org/10.1145/359046.359048

122. Ahuja N, Chien RT, Yen R, Bridwell N (1980) Interference Detection and Collision Avoidance Among Three Dimensional Objects. In: AAAI-80 Proceedings: 44-48.

123. Pan C, Smith SS, Smith GC (2005) Determining interference between parts in CAD STEP files for automatic assembly planning. Journal of Computing and Information Science in Engineering 5:56-62. doi: https://doi.org/10.1115/1.1861473

124. Zubairi MS, Robinson TT, Armstrong CG, Soban DS (2014) A sensitivity approach for eliminating clashes from computer aided design model assemblies. Journal of Computing and Information Science in Engineering 14:031002. doi: https://doi.org/10.1115/1.4027345

125. Immonen E (2017) 2D shape optimization under proximity constraints by CFD and response surface methodology. Appl.Math.Model. 41:508-529. doi: https://doi.org/10.1016/j.apm.2016.09.009

126. Weiss D (2009) Geometry-based structural optimization on CAD specification trees. Dissertation, Swiss Federal Institute of Technology, Zurich